

**PERBANDINGAN JARINGAN SARAF TIRUAN LVQ DENGAN
BACKPROPAGATION DALAM DETEKSI DINI PENYAKIT
JANTUNG KORONER**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Mohammad Setya Adi Fauzi

NIM: 135150201111271



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018

PENGESAHAN

PERBANDINGAN JARINGAN SARAF TIRUAN LVQ DENGAN BACKPROPAGATION
DALAM DETEKSI DINI PENYAKIT JANTUNG KORONER

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh:

Mohammad Setya Adi Fauzi

NIM: 135150201111271

Skripsi ini telah diuji dan dinyatakan lulus pada
30 November 2018

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Bayu Rahayudi, S.T., M.T.
NIP: 19740712 200604 1 001

Candra Dewi, S.Kom., M.Sc.
NIP: 19771114 200312 2 001

Mengetahui

Kaproses Jurusan Teknik Informatika



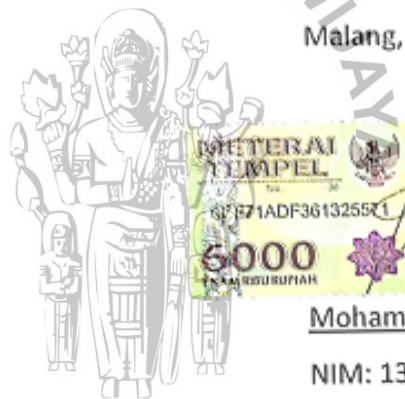
Tri Astoto Kurniawan, S.T., M.T., Ph.D.
NIP: 19710518 200312 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata di dalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 20 Juli 2018



Mohammad Setya Adi Fauzi

NIM: 135150201111271

PRAKATA

Syukur *Alhamdulillah* penulis panjatkan kehadiran Allah SWT yang telah senantiasa melimpahkan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “Perbandingan Jaringan Saraf Tiruan LVQ Dengan *Backpropagation* Dalam Deteksi Dini Penyakit Jantung Koroner”.

Sholawat serta salam semoga tetap tercurahkan kepada suri tauladan terbaik yakni Nabi dan Rasul Besar Muhammad SAW, beserta Keluarganya, Sahabatnya, *Tabi'in*, *Tabiut Tabi'in*, dan para Ulama dan Umatnya yang senantiasa mengikuti tuntunan Beliau. Semoga kelak di akhirat nanti kita mendapatkan syafaat Beliau Rasullullah SAW. *Amiin*.

Dalam penulisan skripsi ini, penulis menyadari tidak terlepas dari berbagai kesulitan dalam menyelesaikannya. Namun, *Alhamdulillah Robbil 'Alamin* berkat kemudahan yang diberikan Allah SWT lewat orang-orang yang dicintai-Nya, akhirnya skripsi ini dapat diselesaikan penulis. Oleh karena itu, penulis ingin mengucapkan terimakasih banyak kepada:

1. Bapak Bayu Rahayudi, S.T., M.T. dan Ibu Candra Dewi, S.Kom., M.Sc. selaku dosen pembimbing skripsi yang telah meluangkan waktu dan dengan sabar membimbing penulis sehingga dapat menyelesaikan skripsi ini,
2. Bapak Wayan Firdaus Mahmudy, S.Si., M.T., Ph.D. selaku Dekan Fakultas Ilmu Komputer Universitas Brawijaya,
3. Bapak Tri Astoto Kurniawan, S.T., M.T., Ph.D. selaku Ketua Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya,
4. Bapak Agus Wahyu Widodo, S.T., M.Cs. selaku Ketua Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya,
5. Seluruh dosen Fakultas Ilmu Komputer Universitas Brawijaya atas kesediaannya membagi ilmu kepada penulis selama masa perkuliahan,
6. Teman-teman LKI Al-Fatih Muslim Drenalin yang telah banyak membantu penulis dan memberikan dukungan dalam menyelesaikan skripsi ini,
7. Yang teristimewa kepada kedua orang tua penulis yang telah membesarkan dan memperjuangkan penulis dengan selalu memberikan doa, dukungan materiil dan moril yang tiada henti hingga penulis dapat menyelesaikan studi di FILKOM UB. Penulis mempersembahkan skripsi ini sebagai salah satu perwujudan rasa tanggung jawab dan terima kasih atas cinta dan pengorbanan yang selama ini telah diberikan kepada penulis,
8. Semua pihak yang telah membantu terselesainya skripsi ini yang tidak dapat penulis sebut satu persatu.

Atas jasa mereka, penulis hanya dapat mendoakan agar amal mereka mendapatkan balasan yang lebih baik serta kesuksesan baik di dunia maupun di akhirat. *Allahuma Aamiin*.

Penulis menyadari bahwa skripsi ini tidak lepas dari kekurangan dan juga kesalahan. Oleh karena itu, kritik dan saran yang membangun dari pembaca sangat diharapkan oleh penulis untuk perbaikan kedepannya. Akhir kata penulis, semoga skripsi ini dapat memberikan manfaat bagi pihak yang menggunakannya.

Malang, 30 Oktober 2018

Penulis

mohammadsetyaadifauzi@gmail.com



ABSTRAK

Mohammad Setya Adi Fauzi, Perbandingan Jaringan Saraf Tiruan LVQ Dengan *Backpropagation* Dalam Deteksi Dini Penyakit Jantung Koroner

Pembimbing: Bayu Rahayudi, S.T., M.T. dan Candra Dewi, S.Kom., M.Sc.

Penyakit jantung koroner merupakan salah satu penyebab kematian tertinggi di dunia. Federasi Jantung Dunia memperkirakan angka kematian akibat penyakit ini di kawasan Asia Tenggara mencapai 1,8 juta kasus pada tahun 2014. Di Indonesia pada tahun 2013 tercatat 883.447 orang terdiagnosa penyakit jantung koroner dengan mayoritas penderita berusia 55-64 tahun dan tingkat kematian akibat penyakit ini cukup tinggi, yakni 45% dari seluruh angka kematian di Indonesia sehingga pendeteksian dini penyakit jantung koroner sangat penting agar resiko penyakit ini dapat diperkecil.

Salah satu teknik *machine learning* yang populer saat ini dan cocok dalam kasus ini adalah jaringan saraf tiruan. Jaringan saraf tiruan merupakan sistem yang terinspirasi dari proses penalaran dalam jaringan saraf manusia. Dalam penelitian ini membandingkan performa jaringan saraf tiruan metode LVQ dan *Backpropagation* untuk deteksi dini penyakit jantung koroner. Variabel-variabel penyakit jantung koroner yang digunakan dalam penelitian ini adalah jenis kelamin, usia, denyut nadi, tekanan darah sistol, kolestrol, gula darah, trigliserida, nyeri dada, sesak nafas, dan batuk. Dari hasil pengujian didapatkan hasil bahwa metode *Backpropagation* lebih baik daripada metode LVQ dengan perbandingan nilai akurasi pelatihan sebesar 95,99097% untuk *Backpropagation* berbanding 66,89659% untuk LVQ dan nilai akurasi pengujian sebesar 68,76034% untuk *Backpropagation* berbanding 54,30313% untuk LVQ.

Kata kunci: klasifikasi, penyakit jantung koroner, jaringan saraf tiruan, *learning vector quantization* (LVQ), *backpropagation*

ABSTRACT

Mohammad Setya Adi Fauzi, *Comparison of LVQ Neural Networks with Backpropagation in Early Detection of Coronary Heart Disease*

Supervisors: Bayu Rahayudi, S.T., M.T. and Candra Dewi, S.Kom., M.Sc.

Coronary heart disease is one of the highest causes of death in the world. The World Heart Federation estimates the number of deaths from this disease in Southeast Asia to reach 1.8 million cases in 2014. In Indonesia in 2013 recorded 883,447 people diagnosed with coronary heart disease with the majority of patients aged 55-64 years and the death rate due to this disease is enough high, ie 45% of all deaths in Indonesia, so early detection of coronary heart disease is very important for the risk of this disease can be minimized.

One of the popular machine learning techniques and fits in this case is the artificial neural network. Artificial neural networks are systems that are inspired by reasoning processes in human neural networks. In this study the authors compared artificial neural network LVQ method and Backpropagation method for early detection of coronary heart disease. The variables of coronary heart disease used in this study were gender, age, pulse, systolic blood pressure, cholesterol, blood sugar, triglycerides, chest pain, shortness of breath, and cough. From the results of this evaluation showed that the Backpropagation method is better than the LVQ method with the comparison of the accuracy value of training of 95,99097% for Backpropagation compared to 66,89659% for LVQ and the accuracy value of testing of 68,76034% for Backpropagation compared to 54,30313% for LVQ.

Keywords: *classification, coronary heart disease, artificial neural network, learning vector quantization (LVQ), backpropagation*

DAFTAR ISI

| | |
|--|-------------|
| PENGESAHAN | ii |
| PERNYATAAN ORISINALITAS | iii |
| PRAKATA | iv |
| ABSTRAK | vi |
| ABSTRACT | vii |
| DAFTAR ISI..... | viii |
| DAFTAR TABEL..... | xiii |
| DAFTAR GAMBAR | xv |
| DAFTAR LAMPIRAN | xvii |
| BAB I PENDAHULUAN | 1 |
| 1.1 Latar Belakang..... | 1 |
| 1.2 Rumusan Masalah..... | 2 |
| 1.3 Tujuan | 3 |
| 1.4 Manfaat..... | 3 |
| 1.5 Batasan Masalah | 3 |
| 1.6 Sistematika Pembahasan | 3 |
| BAB II LANDASAN KEPUSTAKAAN | 5 |
| 2.1 Kajian Pustaka | 5 |
| 2.2 Penyakit Jantung Koroner | 6 |
| 2.2.1 Klasifikasi Penyakit Jantung Koroner | 6 |
| 2.2.2 Penyebab Penyakit Jantung Koroner | 7 |
| 2.3 Jaringan Saraf Tiruan | 7 |
| 2.4 <i>Learning Vector Quantization</i> (LVQ)..... | 8 |
| 2.4.1 Arsitektur Jaringan <i>Learning Vector Quantization</i> (LVQ)..... | 8 |
| 2.4.2 Algoritme Jaringan <i>Learning Vector Quantization</i> (LVQ)..... | 9 |
| 2.5 <i>Backpropagation</i> | 10 |
| 2.5.1 Arsitektur Jaringan <i>Backpropagation</i> | 11 |
| 2.5.2 Fungsi Aktivasi Jaringan <i>Backpropagation</i> | 11 |

| | |
|--|-----------|
| 2.5.3 Algoritme Jaringan <i>Backpropagation</i> | 12 |
| 2.5.4 <i>Root Mean Square Error</i> (RMSE)..... | 14 |
| 2.6 Normalisasi Data..... | 15 |
| 2.7 Pengujian Akurasi | 15 |
| 2.8 <i>Overfitting</i> dan <i>Underfitting</i> | 15 |
| 2.9 <i>K-Fold Cross Validation</i> | 16 |
| BAB III METODOLOGI..... | 17 |
| 3.1 Tipe Penelitian | 17 |
| 3.2 Alur Metodologi..... | 17 |
| 3.3 Studi Pustaka..... | 18 |
| 3.4 Pengumpulan Data | 18 |
| 3.5 Normalisasi Data..... | 18 |
| 3.6 Perancangan Model Jaringan LVQ dan <i>Backpropagation</i> | 19 |
| 3.7 Perancangan Uji Coba Model Jaringan LVQ dan <i>Backpropagation</i> | 19 |
| 3.8 Implementasi Model Jaringan LVQ dan <i>Backpropagation</i> | 19 |
| 3.9 Pengujian Model Jaringan LVQ dan <i>Backpropagation</i> | 19 |
| 3.10 Analisis Hasil Pengujian..... | 19 |
| 3.11 Penarikan Kesimpulan | 19 |
| BAB IV PERANCANGAN | 20 |
| 4.1 Formulasi Permasalahan..... | 20 |
| 4.2 Perancangan Model Jaringan | 20 |
| 4.2.1 LVQ..... | 20 |
| 4.2.2 <i>Backpropagation</i> | 23 |
| 4.3 Diagram Alir | 24 |
| 4.3.1 Normalisasi..... | 26 |
| 4.3.2 Metode LVQ | 27 |
| 4.3.3 Metode <i>Backpropagation</i> | 31 |
| 4.4 Perhitungan Manual | 35 |
| 4.4.1 Normalisasi Data | 35 |
| 4.4.2 Metode LVQ | 37 |

| | |
|--|-----------|
| 4.4.3 Metode <i>Backpropagation</i> | 39 |
| 4.5 Perancangan <i>User Interface</i> | 44 |
| 4.5.1 <i>Form</i> Depan..... | 44 |
| 4.5.2 <i>Form</i> Metode LVQ..... | 45 |
| 4.5.3 <i>Form Testing</i> LVQ..... | 46 |
| 4.5.4 <i>Form</i> Metode <i>Backpropagation</i> | 47 |
| 4.5.5 <i>Form Testing Backpropagation</i> | 48 |
| 4.5.6 <i>Form</i> Hasil Pelatihan | 48 |
| 4.6 Perancangan Uji Coba | 49 |
| 4.6.1 Rancangan Pengujian Parameter <i>Learning Rate</i> | 49 |
| 4.6.2 Rancangan Pengujian Parameter Pengurangan <i>Learning Rate</i> Untuk LVQ | 50 |
| 4.6.3 Rancangan Pengujian Pasangan Data Latih dan Uji | 50 |
| 4.6.4 Rancangan Pengujian Jumlah Neuron Untuk <i>Backpropagation</i> | 50 |
| 4.6.5 Rancangan Pengujian Jumlah <i>Hidden Layer</i> Untuk <i>Backpropagation</i> | 51 |
| 4.6.6 Rancangan Pengujian Jenis Fungsi Aktivasi Untuk <i>Backpropagation</i> | 51 |
| 4.6.7 <i>Cross Validation</i> (CV)..... | 51 |
| BAB V IMPLEMENTASI | 52 |
| 5.1 Implementasi Program | 52 |
| 5.1.1 Implementasi Normalisasi Data | 52 |
| 5.1.2 Implementasi Membuat Jaringan LVQ..... | 54 |
| 5.1.3 Implementasi Pelatihan LVQ..... | 55 |
| 5.1.4 Implementasi Pengujian LVQ..... | 57 |
| 5.1.5 Implementasi Membuat Jaringan <i>Backpropagation</i> | 58 |
| 5.1.6 Implementasi Pelatihan <i>Backpropagation</i> | 59 |
| 5.1.7 Implementasi Pengujian <i>Backpropagation</i> | 63 |
| 5.2 Hasil Implementasi Antarmuka Pengguna..... | 65 |
| 5.2.1 <i>Form</i> Depan | 65 |
| 5.2.2 <i>Form</i> Metode LVQ..... | 67 |

| | |
|--|------------|
| 5.2.3 Form Testing LVQ | 67 |
| 5.2.4 Form Metode Backpropagation | 68 |
| 5.2.5 Form Testing Bacpropagation | 69 |
| 5.2.6 Form Hasil Pelatihan | 69 |
| BAB VI HASIL DAN PEMBAHASAN | 71 |
| 6.1 Analisis Data Deskriptif | 71 |
| 6.2 Hasil dan Pembahasan Pengujian Parameter <i>Learning Rate</i> Untuk Metode LVQ | 72 |
| 6.3 Hasil dan Pembahasan Pengujian Parameter Pengurangan <i>Learning rate</i> Untuk Metode LVQ | 74 |
| 6.4 Hasil dan Pembahasan Pengujian Parameter <i>Learning rate</i> Untuk Metode Backpropagation | 75 |
| 6.5 Hasil dan Pembahasan Pengujian Jumlah Neuron dalam Arsitektur Backpropagation | 77 |
| 6.6 Hasil dan Pembahasan Pengujian Jumlah <i>Hidden Layer</i> dalam Arsitektur Backpropagation | 78 |
| 6.7 Hasil dan Pembahasan Pengujian Arsitektur Backpropagation | 80 |
| 6.8 Hasil dan Pembahasan Pengujian Jenis Fungsi Aktivasi Untuk Metode Backpropagation | 81 |
| 6.9 Hasil dan Pembahasan Pasangan Data Latih dan Uji Untuk Metode LVQ dan Backpropagation | 83 |
| 6.10 Hasil dan Pembahasan Pengujian <i>Cross Validation</i> | 85 |
| BAB VII PENUTUP | 89 |
| 7.1 Kesimpulan | 89 |
| 7.2 Saran | 89 |
| DAFTAR PUSTAKA | 90 |
| LAMPIRAN A DATA 130 PENYAKIT JANTUNG KORONER | 92 |
| LAMPIRAN B GAMBAR SEBARAN DATA BERAGAI PARAMETER PENYAKIT JANTUNG KORONER | 98 |
| LAMPIRAN C HASIL PENGUJIAN | 106 |
| C.1 Hasil Pengujian Parameter <i>Learning rate</i> Untuk Metode LVQ | 106 |

| | |
|--|-----|
| C.2 Hasil Pengujian Parameter <i>Learning rate</i> Untuk Metode <i>Backpropagation</i> | 107 |
| C.3 Hasil Pengujian Parameter Pengurangan <i>Learning rate</i> Untuk Metode LVQ..... | 109 |
| C.4 Hasil Pengujian Jumlah Neuron Untuk Metode <i>Backpropagation</i> | 110 |
| C.5 Hasil Pengujian Jumlah <i>Hidden Layer</i> Untuk Metode <i>Backpropagation</i> | 111 |
| C.6 Hasil Pengujian Arsitektur Metode <i>Backpropagation</i> | 112 |
| C.7 Hasil Pengujian Jenis Fungsi Aktivasi Untuk Metode <i>Backpropagation</i> | 113 |
| C.8 Hasil Pengujian Pasangan Data Latih dan Uji Untuk Metode LVQ | 114 |
| C.9 Hasil Pengujian Pasangan Data Latih dan Uji Untuk Metode <i>Backpropagation</i> | 115 |
| C.10 Hasil Pengujian <i>Cross Validation</i> Untuk Metode LVQ | 116 |
| C.11 Hasil Pengujian <i>Cross Validation</i> Untuk Metode <i>Backpropagation</i> | 117 |



DAFTAR TABEL

| | |
|---|----|
| Tabel 4.1 Sampel data pasien penyakit jantung koroner | 21 |
| Tabel 4.2 Nilai maksimum dan minimum setiap fitur | 36 |
| Tabel 4.3 Hasil normalisasi data | 36 |
| Tabel 4.4 Bobot yang terbentuk | 37 |
| Tabel 4.4 Inisialisasi bobot | 37 |
| Tabel 4.5 Jarak data latih ke-1 terhadap setiap kelas | 38 |
| Tabel 4.6 Jarak data latih ke-1 terhadap setiap kelas | 38 |
| Tabel 4.7 Perbaikan bobot | 38 |
| Tabel 4.8 Bobot akhir | 38 |
| Tabel 4.9 Sample data uji | 39 |
| Tabel 4.10 Hasil <i>testing</i> | 39 |
| Tabel 4.11 Bobot yang terbentuk | 40 |
| Tabel 4.12 Inisialisasi bobot | 40 |
| Tabel 4.13 Hasil signal pada data latih pertama | 41 |
| Tabel 4.14 Hasil koreksi bobot data latih-1 | 41 |
| Tabel 4.15 Hasil perubahan bobot untuk data latih-1 | 42 |
| Tabel 4.16 Bobot akhir | 43 |
| Tabel 4.17 Sample data uji | 43 |
| Tabel 4.18 Hasil uji | 44 |
| Tabel 4.19 Rancangan pengujian parameter <i>learning rate</i> | 49 |
| Tabel 4.20 Rancangan pengujian parameter pengurangan <i>learning rate</i> | 50 |
| Tabel 4.21 Rancangan pengujian pasangan data latih dan uji | 50 |
| Tabel 4.22 Rancangan pengujian jumlah neuron | 50 |
| Tabel 4.23 Rancangan pengujian jumlah <i>hidden layer</i> | 51 |
| Tabel 4.24 Rancangan pengujian jenis fungsi aktivasi | 51 |
| Tabel 4.25 Rancangan pengujian <i>cross validation</i> | 51 |
| Tabel 6.1 Hasil pengujian parameter <i>learning rate</i> dalam metode LVQ dengan peningkatan 0,01 | 72 |

| | | |
|------------|--|----|
| Tabel 6.2 | Hasil pengujian parameter <i>learning rate</i> dalam metode LVQ dengan peningkatan 0,1..... | 72 |
| Tabel 6.3 | Hasil pengujian parameter pengurangan <i>learning rate</i> dalam metode LVQ..... | 74 |
| Tabel 6.4 | Hasil pengujian parameter <i>learning rate</i> dalam metode <i>backpropagation</i> dengan peningkatan 0,01 | 75 |
| Tabel 6.5 | Hasil pengujian parameter <i>learning rate</i> dalam metode <i>Backpropagation</i> dengan peningkatan 0,1 | 76 |
| Tabel 6.6 | Hasil pengujian jumlah neuron dalam metode <i>Backpropagation</i> | 77 |
| Tabel 6.7 | Hasil pengujian jumlah <i>hidden layer</i> dalam metode <i>Backpropagation</i> | 78 |
| Tabel 6.8 | Lama waktu pelatihan setiap jumlah <i>hidden layer</i> dalam metode <i>Backpropagation</i> | 78 |
| Tabel 6.9 | Hasil pengujian arsitektur <i>Backpropagation</i> | 80 |
| Tabel 6.10 | Lama waktu pelatihan setiap arsitektur <i>Backpropagation</i> | 80 |
| Tabel 6.11 | Hasil pengujian jenis fungsi aktivasi setiap arsitektur <i>Backpropagation</i> | 82 |
| Tabel 6.12 | Lama waktu pelatihan setiap arsitektur <i>Backpropagation</i> | 82 |
| Tabel 6.13 | Hasil pengujian pasangan data latih dan uji untuk metode LVQ..... | 83 |
| Tabel 6.14 | Hasil pengujian pasangan data latih dan uji untuk metode <i>Backpropagation</i> | 84 |
| Tabel 6.15 | <i>Fold</i> ke-i dan jumlah datanya | 85 |
| Tabel 6.16 | Model pengujian <i>cross validation</i> | 86 |
| Tabel 6.17 | Hasil pengujian <i>cross validation</i> metode LVQ..... | 86 |
| Tabel 6.18 | Lama Waktu pelatihan pengujian <i>cross validation</i> metode LVQ..... | 86 |
| Tabel 6.19 | Hasil pengujian <i>cross validation</i> metode <i>Backpropagation</i> | 87 |
| Tabel 6.20 | Lama waktu pelatihan pengujian <i>cross validation</i> metode <i>Backpropagation</i> | 87 |
| Tabel 6.21 | Rerata akurasi data latih dan uji metode LVQ dan <i>Backpropagation</i> | 88 |

DAFTAR GAMBAR

| | | |
|-------------|---|----|
| Gambar 2.1 | Arsitektur LVQ | 8 |
| Gambar 2.2 | Arsitektur <i>Backpropagation</i> | 11 |
| Gambar 2.3 | Grafik fungsi aktivasi <i>sigmoid</i> biner | 11 |
| Gambar 2.4 | Grafik fungsi aktivasi <i>sigmoid</i> bipolar | 12 |
| Gambar 2.5 | Kondisi model <i>underfitting</i> , optimal, <i>overfitting</i> | 16 |
| Gambar 4.1 | Model jaringan LVQ..... | 22 |
| Gambar 4.2 | Model jaringan <i>Backpropagation</i> | 23 |
| Gambar 4.3 | Diagram alir program | 25 |
| Gambar 4.4 | Diagram alir proses normalisasi | 27 |
| Gambar 4.5 | Diagram alir proses <i>training</i> LVQ | 29 |
| Gambar 4.6 | Diagram alir proses <i>testing</i> LVQ | 30 |
| Gambar 4.7 | Diagram alir proses <i>training Backpropagation</i> | 33 |
| Gambar 4.8 | Diagram alir proses <i>testing Backpropagation</i> | 35 |
| Gambar 4.9 | <i>Form</i> depan..... | 45 |
| Gambar 4.10 | <i>Form</i> metode LVQ persiapan <i>training</i> | 45 |
| Gambar 4.11 | <i>Form</i> metode LVQ <i>testing</i> | 46 |
| Gambar 4.12 | <i>Form</i> metode <i>Backpropagation</i> | 47 |
| Gambar 4.13 | <i>Form</i> metode <i>Backpropagation testing</i> | 48 |
| Gambar 4.14 | <i>Form</i> hasil pelatihan | 49 |
| Gambar 5.1 | <i>Form</i> depan..... | 66 |
| Gambar 5.2 | <i>Form</i> scan data | 66 |
| Gambar 5.3 | <i>Form</i> metode LVQ | 67 |
| Gambar 5.4 | <i>Form</i> <i>testing</i> LVQ..... | 68 |
| Gambar 5.5 | <i>Form</i> metode <i>Backpropagation</i> | 68 |
| Gambar 5.6 | <i>Form</i> <i>testing Backpropagation</i> | 69 |
| Gambar 5.7 | <i>Form</i> hasil pelatihan LVQ..... | 69 |
| Gambar 5.8 | <i>Form</i> hasil pelatihan <i>Backpropagation</i> | 70 |
| Gambar 6.1 | Diagram jumlah data setiap kelas | 71 |

| | | |
|-------------|--|----|
| Gambar 6.2 | <i>Scatter plot</i> kolesterol dengan trigliserida | 71 |
| Gambar 6.3 | Grafik peningkatan <i>learning rate</i> dengan peningkatan 0,01 | 73 |
| Gambar 6.4 | Grafik peningkatan <i>learning rate</i> dengan peningkatan 0,1 | 73 |
| Gambar 6.5 | Grafik pengurangan <i>learning rate</i> (0.01 sampai 0.1) | 75 |
| Gambar 6.6 | Grafik peningkatan <i>learning rate</i> dengan peningkatan 0,01 | 76 |
| Gambar 6.7 | Grafik peningkatan <i>learning rate</i> dengan peningkatan 0,1 | 77 |
| Gambar 6.8 | Grafik peningkatan jumlah neuron | 78 |
| Gambar 6.9 | Grafik peningkatan jumlah <i>hidden layer</i> | 79 |
| Gambar 6.10 | Grafik lama waktu pelatihan setiap <i>hidden layer</i> | 79 |
| Gambar 6.11 | Grafik hasil pengujian arsitektur <i>Backpropagation</i> | 81 |
| Gambar 6.12 | Grafik lama waktu pelatihan setiap arsitektur <i>Backpropagation</i> . | 81 |
| Gambar 6.13 | Grafik hasil pengujian fungsi aktivasi setiap arsitektur <i>Backpropagation</i> | 82 |
| Gambar 6.14 | Grafik lama waktu pelatihan setiap arsitektur <i>Backpropagation</i> . | 83 |
| Gambar 6.15 | Grafik hasil pengujian pasangan data latih dan uji untuk metode LVQ | 84 |
| Gambar 6.16 | Grafik hasil pengujian pasangan data latih dan uji untuk metode <i>Backpropagation</i> | 85 |

DAFTAR LAMPIRAN

| | |
|---|-----|
| LAMPIRAN A DATA 130 PENYAKIT JANTUNG KORONER | 92 |
| LAMPIRAN B GAMBAR SEBARAN DATA BERAGAI PARAMETER PENYAKIT JANTUNG KORONER..... | 98 |
| LAMPIRAN C HASIL PENGUJIAN | 106 |
| C.1 Hasil Pengujian Parameter <i>Learning Rate</i> Untuk Metode LVQ..... | 106 |
| C.2 Hasil Pengujian Parameter <i>Learning Rate</i> Untuk Metode <i>Backpropagation</i> | 107 |
| C.3 Hasil Pengujian Parameter Pengurangan <i>Learning Rate</i> Untuk Metode LVQ..... | 109 |
| C.4 Hasil Pengujian Jumlah Neuron Untuk Metode <i>Backpropagation</i> | 110 |
| C.5 Hasil Pengujian Jumlah <i>Hidden Layer</i> Untuk Metode <i>Backpropagation</i> | 111 |
| C.6 Hasil Pengujian Arsitektur Metode <i>Backpropagation</i> | 112 |
| C.7 Hasil Pengujian Jenis Fungsi Aktivasi Untuk Metode <i>Backpropagation</i> | 113 |
| C.8 Hasil Pengujian Pasangan Data Latih dan Uji Untuk Metode LVQ | 114 |
| C.9 Hasil Pengujian Pasangan Data Latih dan Uji Untuk Metode <i>Backpropagation</i> | 115 |
| C.10 Hasil Pengujian <i>Cross Validation</i> Untuk Metode LVQ..... | 116 |
| C.11 Hasil Pengujian <i>Cross Validation</i> Untuk Metode <i>Backpropagation</i> | 117 |

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Penyakit Jantung Koroner menurut Kertohoesodo (1982) adalah penyakit jantung akibat penyempitan pembuluh nadi koroner yang dapat mengganggu aliran darah ke otot jantung sehingga menyebabkan rasa nyeri di dada. Dalam kondisi lebih parah kemampuan jantung untuk memompa darah dapat terhenti sehingga menyebabkan kematian.

Penyakit jantung koroner merupakan salah satu penyebab kematian tertinggi di dunia. Federasi Jantung Dunia memperkirakan angka kematian akibat penyakit ini di kawasan Asia Tenggara mencapai 1,8 juta kasus pada tahun 2014. Di Indonesia pada tahun 2013 tercatat 883.447 orang terdiagnosa penyakit jantung koroner dengan mayoritas penderita berusia 55-64 tahun dan tingkat kematian akibat penyakit ini cukup tinggi, yakni 45% dari seluruh angka kematian di Indonesia (Aladokter.com, 2017).

Pendeteksian dini penyakit jantung koroner sangat penting agar resiko penyakit ini dapat diperkecil. Biasanya untuk mendeteksi penyakit jantung koroner digunakan Pemeriksaan Tes Darah, Pemeriksaan *Elektrokardiogram*, Pemeriksaan *Ekokardiogram*, Pemeriksaan *X-Ray*, *Angiografi* Koroner atau *Kateterisasi* Jantung, CT dan MRI *Scan* (Aladokter.com, 2017). Namun, dibutuhkan biaya yang relatif mahal dan memerlukan waktu yang tidak sebentar dengan cara tersebut. Sekarang banyak alat kesehatan yang dapat mengukur kondisi darah seseorang dengan cepat dan akurat dengan harga yang relatif terjangkau. Dengan memanfaatkan alat tersebut kita dapat mengetahui kadar kolesterol, gula darah, asam urat, tekanan darah yang dapat dijadikan variabel dalam pendeteksian dini penyakit jantung koroner. Agar mempermudah pendeteksian dini penyakit jantung koroner dibutuhkan suatu sistem deteksi dini penyakit jantung koroner dengan metode klasifikasi yang efektif.

Terdapat banyak penelitian mengenai metode klasifikasi untuk deteksi dini penyakit jantung koroner, antara lain: penelitian yang dilakukan oleh Wardhani (2014) yang berhasil mendeteksi penyakit jantung koroner dengan tingkat keakuratan sebesar 95% menggunakan metode *fuzzy* Mamdani dengan *defuzzifikasi* metode *centroid* berdasarkan 70 data latih dan 20 data uji yang memiliki sebelas variabel, penelitian oleh Effendy dan Faisal (2008) dengan menggunakan jaringan saraf *Backpropagation* MATLAB yang diperoleh tingkat akurasi sebesar 80% dengan 40 data latih yang memiliki sembilan variabel.

Memang sudah banyak penelitian yang dilakukan dalam pendeteksian dini penyakit jantung koroner. Namun, masih diperlukan peningkatan dalam pengembangannya dari segi penambahan atau pengurangan jumlah variabel, kombinasi variabel yang digunakan, dan perbandingan berbagai metode

klasifikasi yang digunakan untuk mencapai tingkat keakuratan deteksi yang optimal.

Salah satu teknik *machine learning* yang dapat diterapkan dalam kasus ini adalah jaringan saraf tiruan karena hampir 90% semua kasus klasifikasi dapat diselesaikan dengan jaringan saraf tiruan (Utami dan Ulama, 2015). Jaringan saraf tiruan merupakan sistem yang terinspirasi dari proses penalaran dalam jaringan saraf manusia. Menurut Kusumadewi (2003) terdapat dua jenis metode pembelajaran pada jaringan saraf tiruan, yaitu pembelajaran terawasi yang membutuhkan data latih (*supervised learning*) dan pembelajaran tak terawasi yang tidak membutuhkan data latih (*unsupervised learning*). Dalam proses deteksi penyakit jantung koroner pembelajaran terawasi lebih cocok digunakan karena proses tersebut membutuhkan data latih dari sumber yang valid agar dapat menghasilkan deteksi yang akurat.

Pembelajaran LVQ dan *Backpropagation* merupakan metode pembelajaran pada jaringan saraf tiruan yang terawasi. Pembelajaran LVQ cukup populer dalam proses klasifikasi suatu dataset dan mampu menghasilkan tingkat keakuratan yang baik. Hal itu dapat dibuktikan dari banyaknya penelitian klasifikasi yang menggunakan LVQ, antara lain: penelitian oleh Megawati dan Candra (2017) mengenai diagnosis hama dan penyakit jeruk yang menghasilkan akurasi sebesar 93,10%, dan Hamidi, et al. (2017) dapat menghasilkan akurasi sebesar 81,13% dalam klasifikasi kualitas air sungai. Pembelajaran *Backpropagation* juga menjadi salah satu metode pembelajaran jaringan saraf tiruan yang sering diterapkan dalam penelitian, antara lain: Luthfie (2007) berhasil mendapatkan akurasi 82.23% untuk pengenalan wajah, dan Lesnussa, et al. (2013) menghasilkan tingkat akurasi 90% dalam sistem pengenalan *barcode*.

Dari penjelasan yang telah dipaparkan, saya berkeinginan untuk menulis skripsi mengenai perbandingan metode jaringan saraf tiruan pembelajaran LVQ dengan *Backpropagation* untuk mendeteksi penyakit jantung koroner. Pengerjaan skripsi ini diharapkan dapat memberikan penjelasan mengenai perbandingan jaringan saraf tiruan LVQ dengan *Backpropagation* untuk deteksi penyakit jantung koroner.

1.2 Rumusan Masalah

Berdasarkan latar belakang tersebut terbentuk rumusan masalah yang menjadi dasar penelitian skripsi yaitu:

1. Apa saja parameter yang digunakan agar deteksi penyakit jantung koroner LVQ dan *Backpropagation* memperoleh hasil yang optimal?
2. Bagaimana perbandingan hasil deteksi penyakit jantung koroner antara LVQ dan *Backpropagation*?
3. Manakah yang lebih efektif antara LVQ atau *Backpropagation* untuk deteksi penyakit jantung koroner?

1.3 Tujuan

Tujuan yang ingin dicapai dari penelitian skripsi ini adalah:

1. Mengetahui parameter-parameter yang digunakan untuk memperoleh hasil optimal dalam deteksi penyakit jantung koroner antara LVQ dan *Backpropagation*
2. Mengetahui hasil perbandingan deteksi penyakit jantung koroner antara LVQ dan *Backpropagation*
3. Memberikan rekomendasi metode yang lebih baik antara LVQ dan *Backpropagation* untuk deteksi penyakit jantung koroner.

1.4 Manfaat

Manfaat dari penelitian skripsi ini adalah:

1. Membantu pihak terkait seperti rumah tangga ataupun tenaga medis dalam mendeteksi penyakit jantung koroner
2. Memberikan informasi baru mengenai perbandingan LVQ dan *Backpropagation* bagi peneliti yang meminati jaringan saraf tiruan
3. Menambah khazanah penelitian mengenai metode pembelajaran jaringan saraf tiruan LVQ dan *Backpropagation* di Fakultas Ilmu Komputer Universitas Brawijaya.

1.5 Batasan Masalah

Terdapat beberapa batasan dalam penelitian ini:

1. Data yang dipakai adalah data sekunder dari penelitian mengenai penyakit jantung koroner dengan fitur sebanyak 10 macam, yakni: jenis kelamin, usia, denyut nadi, tekanan darah sistol, kolestrol, gula darah, trigliserida, nyeri dada, sesak nafas, dan batuk
2. Parameter pengujian yang digunakan adalah tingkat keakuratan dalam proses pelatihan dan pengujian, serta lama waktu pelatihan.

1.6 Sistematika Pembahasan

Untuk memberikan gambaran secara menyeluruh mengenai skripsi ini, maka sistematika penulisan dibagi dalam tujuh bab sebagai berikut.

1. BAB I PENDAHULUAN

Pada bab ini dibahas mengenai latar belakang penelitian, rumusan masalah, tujuan, manfaat, batasan masalah, dan sistematika penulisan skripsi ini.

2. BAB II LANDASAN KEPUSTAKAAN

Bab ini berisi tentang landasan teori yang berkaitan dengan penyakit jantung koroner dan metode pembelajaran jaringan saraf tiruan LVQ dan *Backpropagation*.

3. BAB III METODOLOGI

Bab ini menjelaskan tentang metodologi penelitian yang akan digunakan. Metodologi penelitian ini dapat memberikan gambaran secara menyeluruh tentang tahap-tahap yang akan dilakukan dalam penelitian.

4. BAB IV PERANCANGAN

Bab ini menjelaskan tentang perancangan aplikasi, mulai dari manualisasi metode sampai perancangan UI dan rancangan pengujian.

5. BAB V IMPLEMENTASI

Bab ini menjelaskan tentang penerapan rancangan aplikasi yang telah dijelaskan dalam bab sebelumnya.

6. BAB VI HASIL DAN PEMBAHASAN

Bab ini memuat uji coba aplikasi yang telah diimplementasi dan memastikan kinerja aplikasi sudah sesuai dengan tujuan yang telah dirumuskan.

7. BAB VII PENUTUP

Bab ini berisi tentang kesimpulan dari penelitian dan saran mengenai penelitian yang telah dilakukan.

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Kajian pustaka pada penelitian ini membahas tentang penelitian yang sebelumnya telah dilakukan dengan metode yang sama tetapi dengan objek yang berbeda atau dengan objek yang sama tetapi dengan metode yang berbeda. Objek penelitian ini adalah data pasien penyakit jantung koroner dan pasien sehat, yang akan digunakan untuk membandingkan metode klasifikasi jaringan saraf tiruan antara jaringan saraf tiruan pembelajaran LVQ dengan pembelajaran *Backpropagation*.

Penelitian pertama yang dilakukan oleh Hawickhorst dan Zahorian (1996). Dalam penelitian tersebut Hawickhorst dan Zahorian mencoba untuk membandingkan tiga arsitektur jaringan saraf tiruan yaitu *Backpropagation*, *Radial Basis Function* (RBF), dan *Learning Vector Quantization* (LVQ) dalam pengenalan suara digital. Penelitian tersebut menghasilkan suatu kesimpulan bahwa metode RBF dan LVQ lebih baik daripada *Backpropagation* dalam proses efisiensi klasifikasi pengenalan suara namun tingkat akurasi *Backpropagation* sedikit lebih baik daripada metode RBF dan LVQ dengan waktu komputasi pembelajaran yang lama.

Penelitian kedua, Effendy dan Faisal (2008) mengenai prediksi penyakit jantung koroner berdasarkan faktor risiko menggunakan jaringan saraf tiruan *backpropagation*. Penelitian tersebut bertujuan untuk memprediksi penyakit jantung koroner (sehat atau sakit) dengan pendekatan *machine learning* jaringan saraf tiruan *Backpropagation* yang disimulasikan dengan MATLAB 7.0.4. Penelitian tersebut menggunakan sembilan faktor risiko penyebab timbulnya penyakit jantung koroner sebagai variabel *input* yakni: Usia (dalam tahun), Jenis kelamin, Pekerjaan, Kadar LDL, Kadar kolesterol total, Kadar HDL, Kadar trigliserida, Tekanan darah sistolik, Tekanan darah diastolik. Hasil pengujian dari penelitian tersebut adalah 80% untuk data uji sebanyak 20 data dengan 40 data latih.

Penelitian ketiga, Wardhani (2014) mengenai aplikasi sistem *fuzzy* untuk diagnosis penyakit jantung koroner. Penelitian tersebut bertujuan untuk mendiagnosis jenis penyakit jantung koroner yakni PJK tingkat 1 (*Asimtomatik*), PJK tingkat 2 (*Angina Pectoris*), PJK tingkat 3 (*Infark Miokard Akut/ IMA*). Variabel *input* yang digunakan yaitu jenis kelamin, usia, denyut nadi, tekanan darah sistolik, kolesterol, gula darah sewaktu, trigliserida, tipe *elektrokardiogram* (EKG), nyeri dada, sesak nafas, dan batuk. Sistem inferensi *fuzzy* yang digunakan dalam penelitian tersebut yaitu sistem inferensi Mamdani yang menggunakan *defuzzifikasi centroid* dan MOM. Hasil dari penelitian tersebut diperoleh tingkat keakuratan pada metode *defuzzifikasi centroid* sebesar 98,5% untuk data latih dan

95% untuk data uji, sedangkan untuk metode *defuzzifikasi* MOM diperoleh tingkat keakuratan data latih sebesar 98,5% dan data uji sebesar 90%.

Penelitian keempat, mengenai perbandingan jaringan saraf tiruan *Backpropagation* dengan LVQ oleh Leleury, Lesnussa dan Madiuw (2016). Penelitian ini memiliki tujuan untuk melakukan diagnosis penyakit dalam pada manusia. Hasil cukup mengejutkan, perbedaan tingkat akurasi antara *Backpropagation* dan LVQ sangatlah besar yakni 61,84% berbanding 93,42%. Hal tersebut terjadi karena arsitektur jaringan saraf tiruan *Backpropagation* dalam penelitian tersebut mungkin terlalu sederhana karena dalam jurnal tersebut tidak diterangkan arsitektur jaringan saraf tiruan *Backpropagation* yang digunakan.

Penelitian kelima, Megawati dan Candra (2017) mengenai diagnosis hama dan penyakit tanaman jeruk. Mereka menggunakan jaringan saraf tiruan pembelajaran LVQ untuk mengklasifikasi data 21 gejala hama dan 34 penyakit tanaman jeruk ke dalam 12 kategori kelas untuk hama dan 17 kategori kelas untuk penyakit. Penelitian ini didapatkan tingkat akurasi yang sangat tinggi yang mencapai 93,10%.

2.2 Penyakit Jantung Koroner

Penyakit Jantung Koroner menurut Kertohoesodo (1982) adalah penyakit jantung akibat penyempitan pembuluh nadi koroner yang dapat mengganggu aliran darah ke otot jantung sehingga menyebabkan rasa nyeri di dada. Dalam kondisi lebih parah kemampuan jantung untuk memompa darah dapat terhenti sehingga menyebabkan kematian.

2.2.1 Klasifikasi Penyakit Jantung Koroner

Terdapat tiga klasifikasi penyakit jantung koroner (Gray, et al., 2002 disitasi dalam Wardhani, 2014), yaitu:

a. *Asimtomatik (Silent Myocardial Ischemia)*

Pada tingkat penyakit jantung koroner ini, penderita tidak mengalami keluhan nyeri di dada (angina) baik saat beristirahat dan beraktivitas. Namun, ketika dilakukan pemeriksaan ternyata terdapat gangguan pada area jantung.

b. *Angina Pektoris*

1) *Angina Pektoris Stabil (STEMI)*

Terdapat nyeri dada saat aktivitas berlangsung selama 1-5 menit dan hilang saat istirahat. Nyeri dada bersifat kronik (>2 bulan). Nyeri terutama di daerah dada kiri, terasa seperti tertekan benda berat atau terasa panas dan menjalar ke lengan kiri, leher, dagu, punggung, dan jarang menjalar pada lengan kanan.

2) Angina Pectoris tidak stabil (NSTEMI)

Ciri penderi hamipr sama dengan STEMI tetapi nyeri lebih bersifat progresif dengan frekuensi yang meningkat dan sering terjadi saat istirahat.

c. Infark Miokard Akut (IMA)

Sering didahului dada terasa tidak enak (chest discomfort). Nyeri dada seperti tertekan, teremas, tercekik, berat, tajam dan terasa panas, berlangsung >30 menit bahkan sampai berjam-jam. Pemeriksaan fisik didapatkan pasien tampak ketakutan, gelisah, tegang, dan nadi sering menurun.

2.2.2 Penyebab Penyakit Jantung Koroner

Terdapat beberapa faktor resiko penyebab penyakit jantung koroner. Secara garis besar faktor tersebut dapat dibagi menjadi 2, yaitu faktor resiko yang dapat diubah dan faktor resiko yang tidak dapat diubah (Wardhani, 2014).

a. Faktor Resiko yang Dapat Diubah

1. Kolesterol
2. Hipertensi
3. Merokok
4. Obesitas
5. Diabetes Melitus
6. Kurang Aktivitas Fisik
7. Stres

b. Faktor Resiko yang Tidak Dapat Diubah

1. Genetik
2. Jenis Kelamin
3. Usia

2.3 Jaringan Saraf Tiruan

Jaringan saraf tiruan merupakan sistem informasi yang diproses dengan karakteristik seperti saraf biologis manusia (Fausett, 1994). Jaringan saraf tiruan telah dikembangkan sebagai generalisasi dari model matematika dari kesadaran (kognisi) manusia atau jaringan biologi, berdasarkan asumsi-asumsi bahwa:

1. Pemrosesan informasi terjadi pada beberapa elemen sederhana disebut neuron
2. Signal yang lewat antar neuron melalui penghubung (link koneksi)
3. Setiap link koneksi memiliki bobot yang sesuai, yang mana pada tipe saraf tertentu, perkalian signal ditransmisikan

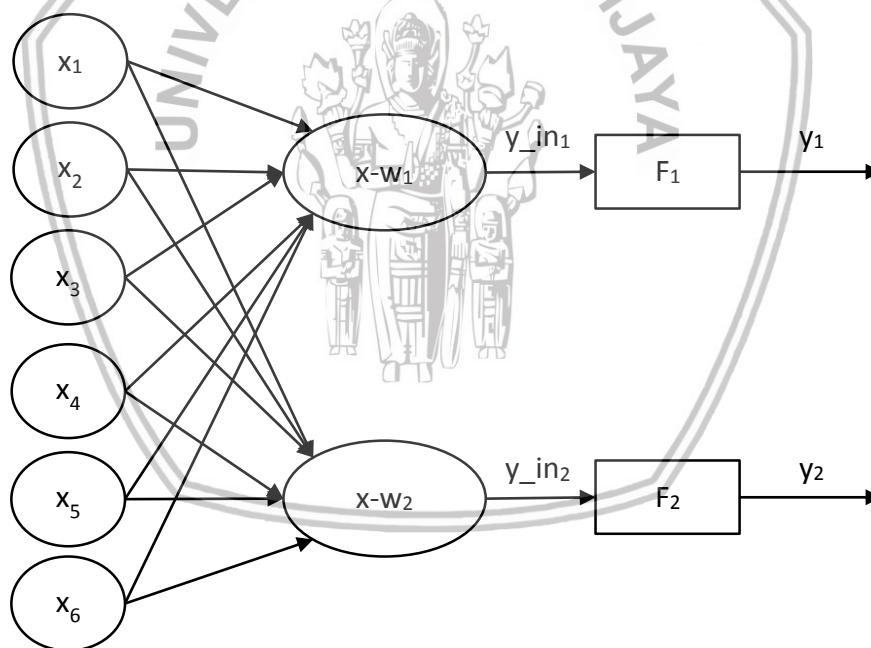
- Setiap neuron menerapkan fungsi aktivasi (umumnya non linier) kepada *input* jaringan (penjumlahan bobot signal *input*) untuk menentukan signal *output*nya.

2.4 Learning Vector Quantization (LVQ)

Jaringan LVQ adalah jaringan yang mengklasifikasi pola sehingga setiap unit keluaran menyatakan suatu kelas (Widodo, 2013). Vektor bobot untuk unit keluaran sering disebut vektor referensi (buku kode) untuk kelas yang dinyatakan oleh unit tersebut dimana vektor bobot tersebut akan diatur bobotnya lewat pelatihan terawasi.

LVQ melakukan pembelajaran pada lapisan kompetitif yang terawasi (Kusumadewi, 2003). Suatu lapisan kompetitif akan secara otomatis belajar untuk mengklasifikasikan vektor *input* yang diberikan. Jika beberapa vektor *input* memiliki jarak yang berdekatan, maka vektor-vektor *input* tersebut akan dikelompokkan ke dalam kelas yang sama.

2.4.1 Arsitektur Jaringan Learning Vector Quantization (LVQ)



Gambar 2.1 Arsitektur LVQ

Gambar 2.1 menunjukkan jaringan LVQ dengan enam unit (neuron) dalam lapisan *input* dan dua unit dalam lapisan *output*. Pemrosesan yang terjadi di setiap unit (neuron) adalah mencari jarak antara suatu vektor *input* ke bobot yang bersangkutan (w_1 atau w_2). w_1 adalah vektor bobot yang menghubungkan setiap neuron pada lapisan *input* ke neuron pertama pada lapisan *output* sedangkan w_2 adalah vektor bobot yang menghubungkan setiap neuron pada lapisan *input* ke neuron kedua pada lapisan *output*. Fungsi aktivasi F_1 akan memetakan y_{in1} ke

$y_1 = 1$ apabila $|x-w_1| < |x-w_2|$, dan $y_1 = 0$ jika sebaliknya. Demikian pula dengan yang terjadi dalam fungsi aktivasi F_2 , akan memetakan y_{in2} ke $y_2 = 1$ apabila $|x-w_2| < |x-w_1|$, dan $y_2 = 0$ jika sebaliknya.

2.4.2 Algoritme Jaringan *Learning Vector Quantization* (LVQ)

Dalam bukunya Kusumadewi (2004) algoritme jaringan LVQ dibedakan menjadi dua, yakni algoritme pelatihan dan algoritme simulasi (pengujian). Misalkan terdapat n buah data dengan m buah variabel *input*. Data tersebut dikelompokkan dalam K kelas.

Algoritme Pelatihnannya,

Langkah 0, Tetapkan:

1. Bobot awal variabel *input* ke- j menuju ke kelas ke- i : w_{ij} dengan $i = 1, 2, \dots, K$; dan $j = 1, 2, \dots, m$,
2. Maksimum *epoch* atau *MaxEpoch*,
3. Parameter *learning rate* α ,
4. Parameter pengurangan *learning rate*, *Deca*,
5. Minimal *learning rate* yang diperbolehkan, *Min α* .

Langkah 1, Masukkan:

1. Data *input*: X_{ij} ; dengan $i = 1, 2, \dots, n$ dan $j = 1, 2, \dots, m$,
2. Target berupa kelas: T_k ; dengan $k = 1, 2, \dots, n$.

Langkah 2, tetapkan kondisi awal *epoch* = 0.

Langkah 3, kerjakan jika: (*epoch* \leq *MaxEpoch*) dan ($\alpha \geq$ *Min α*)

1. *Epoch* = *epoch* + 1,
2. Kerjakan untuk $i = 1$ sampai n
 - a. Tentukan j sedemikian hingga $|x_i - w_j|$ minimum menggunakan Persamaan 2.1; dengan $j = 1, 2, \dots, K$,
$$\text{jarak euclidean} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + \dots + (z_1 - z_2)^2} \quad (2.1)$$
 - b. Perbaiki w_j pemenang dengan ketentuan:
 - Jika $T = C_j$ maka $w_j = w_j + \alpha(x_i - w_j)$ (2.2)
 - Jika $T \neq C_j$ maka $w_j = w_j - \alpha(x_i - w_j)$ (2.3)
 - c. Kurangi nilai α , dengan $\alpha = \alpha - \alpha * \text{Deca}$. (2.4)

Setelah dilakukan proses pelatihan akan diperoleh bobot-bobot akhir (w) yang akan digunakan untuk melakukan proses simulasi atau pengujian. Misalkan kita akan menguji n_p buah data.

Algoritme Simulasi (Pengujian):

1. Masukkan data yang akan diuji, misal: X_{ij} dengan $i = 1, 2, \dots, np$ dan $j = 1, 2, \dots, m$,
2. Kerjakan untuk $i = 1$ sampai np
 - a. Tentukan j sedemikian hingga $|x_i - w_j|$ minimum menggunakan Persamaan 2.1; dengan $j = 1, 2, \dots, K$,
 - b. j adalah kelas untuk X_i .

2.5 Backpropagation

Jaringan Saraf Tiruan *Backpropagation* merupakan suatu algoritme pembelajaran yang terawasi, biasanya digunakan oleh *perceptron* dengan banyak lapisan. *Backpropagation* mempunyai cara kerja dengan mengubah nilai bobot-bobot dalam jaringan dengan eror *output* dalam arah mundur (*backward*). Untuk mendapat eror tersebut, tahap perambatan maju (*forward propagation*) harus dilakukan terlebih dahulu (Kusumadewi, 2004).

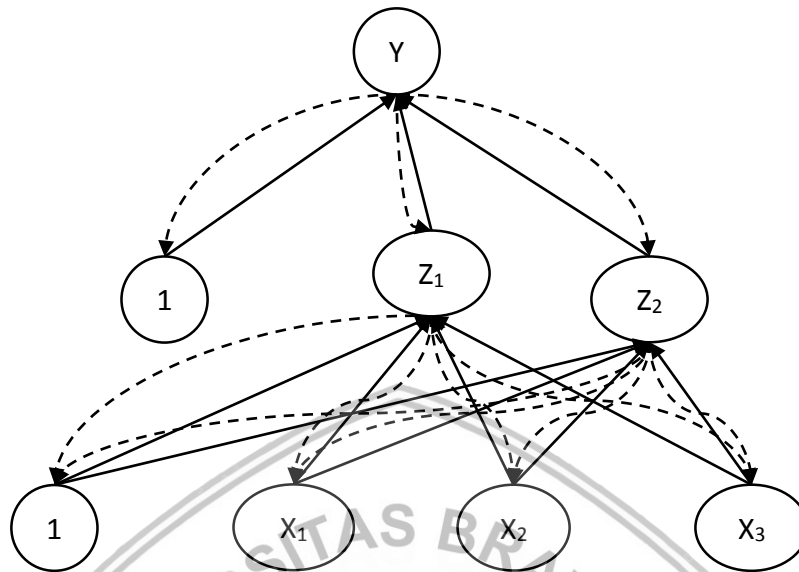
Menurut Widodo (2013), pelatihan jaringan *Backpropagation* melibatkan tiga hal yaitu:

1. Perambatan maju pola pelatihan masukkan,
2. Perhitungan dan perambatan balik galat terkait,
3. Pengaturan bobot.

Setelah pelatihan, jaringan hanya melakukan komputasi perambatan maju. Dalam proses pelatihan jaringan *Backpropagation* memang memerlukan waktu yang relatif lama tetapi proses pengujian atau simulasinya relatif cepat.

Metode *Backpropagation* banyak diaplikasikan secara luas. Sekitar 90% bahkan lebih (Utami dan Ulama, 2015), *Backpropagation* telah berhasil diaplikasikan di berbagai bidang, yakni bidang finansial, pengenalan pola tulisan tangan, pengenalan pola suara, sistem kendali, pengolahan citra medis dan masih banyak lagi sehingga *Backpropagation* dapat dikatakan sebagai metode komputasi yang handal.

2.5.1 Arsitektur Jaringan *Backpropagation*

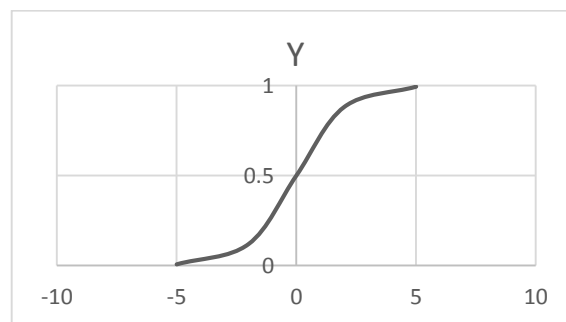


Gambar 2.2 Arsitektur *Backpropagation*

Arsitektur jaringan *backpropagation* pada Gambar 2.2 terdiri atas tiga unit neuron dalam lapisan *input*, yaitu X_1 , X_2 , X_3 ; satu lapisan tersembunyi dengan dua unit neuron, yaitu Z_1 dan Z_2 ; serta satu unit dalam lapisan *output*, yaitu Y . Setiap unit neuron dihubungkan dengan garis lurus tegas yang merepresentasikan bobot antara neuron yang saling terhubung. Garis putus-putus merepresentasikan perambatan galat eror antar neuron yang saling terhubung. Unit neuron bernilai 1 merupakan bias dari lapisan *input* dan lapisan tersembunyi.

2.5.2 Fungsi Aktivasi Jaringan *Backpropagation*

Fungsi aktivasi jaringan *Backpropagation* mempunyai syarat sebagai berikut. kontinyu, terdeferensialkan, naik monoton, derevatifnya mudah dihitung (Widodo, 2013). Fungsi aktivasi yang memenuhi syarat tersebut adalah fungsi *sigmoid* biner dan fungsi *sigmoid* bipolar.



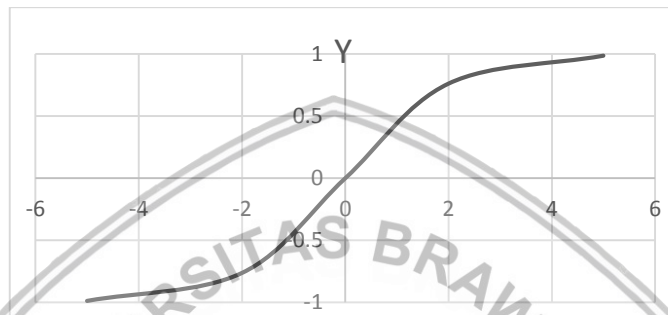
Gambar 2.3 Grafik fungsi aktivasi *sigmoid* biner

Fungsi *sigmoid* biner memiliki nilai pada range 0 sampai 1 sehingga fungsi ini sering digunakan dalam jaringan saraf yang membutuhkan nilai *output* yang terletak pada interval 0 sampai 1 dan nilai *output* yang bernilai 0 atau 1.

Fungsi *sigmoid* biner dirumuskan sebagai berikut.

$$y = \frac{1}{1+e^{-\alpha x}} \quad (2.5)$$

$$\text{Dengan } y' = \alpha y (1 - y) \quad (2.6)$$



Gambar 2.4 Grafik fungsi aktivasi *sigmoid* bipolar

Fungsi *sigmoid* bipolar hampir sama dengan fungsi *sigmoid* biner hanya saja nilai dari fungsi ini berkisar antara -1 sampai 1.

Fungsi *sigmoid* bipolar dirumuskan sebagai berikut.

$$y = \frac{1-e^{-x}}{1+e^{-x}} \quad (2.7)$$

$$\text{Dengan } y' = \frac{\alpha}{2} (1 + y) (1 - y). \quad (2.8)$$

2.5.3 Algoritme Jaringan *Backpropagation*

Algoritme dalam jaringan *Backpropagation* (Kusumadewi, 2004)

1. Inisialisasi bobot (ambil bobot awal dengan nilai *random* dengan nilai antara -0.5 sampai 0.5 atau -1 sampai 1);
2. Tetapkan *MaxEpoch*, *Target Error*, dan *learning rate* (α);
3. Inisialisasi *Epoch* = 0, dan *MSE* = 1;
4. Kerjakan langkah berikut selama (*Epoch* < *MaxEpoch*) dan (*MSE* > *Target Error*);
 - a. *Epoch* = *Epoch* + 1;
 - b. Untuk setiap pasangan elemen yang akan dilakukan pembelajaran, lakukan:

Feedforward

- i. Setiap unit *input* (X_i , $i = 1, 2, 3, \dots, n$) menerima sinyal x_i dan meneruskan sinyal tersebut ke semua unit pada lapisan di atasnya (lapisan tersembunyi);
- ii. Setiap unit pada suatu lapisan tersembunyi (Z_j , $j = 1, 2, 3, \dots, p$) menjumlahkan sinyal-sinyal *input* terbobot.

$$z_in_j = b1_j + \sum_{i=1}^n x_i \times bobot_{ij} \quad (2.9)$$

Gunakan sinyal aktivasi untuk menghitung sinyal *output*nya:

$$z_j = f(z_in_j) \quad (2.10)$$

Lalu kirimkan sinyal tersebut ke semua unit pada lapisan di atasnya (unit-unit lapisan *output*);

- iii. Setiap unit *output* menjumlahkan sinyal-sinyal terbobot yang masuk:

$$y_in_k = b2_k + \sum_{j=1}^p z_j \times bobot_{jk} \quad (2.11)$$

Gunakan sinyal aktivasi untuk menghitung sinyal *output*nya:

$$y_k = f(y_in_k) \quad (2.12)$$

Lalu kirimkan sinyal tersebut ke semua unit pada lapisan di atasnya (unit-unit lapisan *output*);

Catatan langkah ii dilakukan sebanyak jumlah lapisan tersembunyi.

Backpropagation

- iv. Setiap unit *output* menerima target pola yang berhubungan dengan pola *input* pembelajaran, hitung informasi erornya:

$$\delta 2_k = (t_k - y_k) f'(y_in_k) \quad (2.13)$$

$$\omega 2_{jk} = \delta 2_k z_j \quad (2.14)$$

$$\beta 2_k = \delta 2_k \quad (2.15)$$

Hitung koreksi bobot yang akan digunakan untuk perbaikan bobot.

$$\Delta w_{jk} = \alpha \omega 2_{jk} \quad (2.16)$$

Hitung juga koreksi bias

$$\Delta b 2_k = \alpha \beta 2_k \quad (2.17)$$

Langkah iv dilakukan sebanyak jumlah lapisan tersembunyi di bawahnya;

- v. Setiap unit tersembunyi menjumlahkan delta *input*nya dari unit-unit lapisan di atasnya:

$$\delta_in_j = \sum_{k=1}^m \delta 2_k w_{jk} \quad (2.18)$$

Kalikan nilai tersebut dengan turunan dari fungsi aktivasinya untuk menghitung informasi eror:

$$\delta 1_j = \delta_{in_j} f'(z_{in_j}) \quad (2.19)$$

$$\omega 1_{ij} = \delta 1_j x_j \quad (2.20)$$

$$\beta 1_j = \delta 1_j \quad (2.21)$$

Hitung koreksi bobot yang akan digunakan untuk perbaikan bobot.

$$\Delta v_{ij} = \alpha \omega 1_{ij} \quad (2.22)$$

Hitung juga koreksi bias

$$\Delta b 1_j = \alpha \beta 1_j \quad (2.23)$$

- vi. Setiap unit *output*-k ($k = 1, 2, \dots, m$) memperbaiki bobot-j ($j = 1, 2, \dots, p$) dan biasnya;

$$w_{jk} \text{baru} = w_{jk} \text{lama} + \Delta w_{jk} \quad (2.24)$$

$$b 2_k \text{baru} = b 2_k \text{lama} + \Delta b 2_k \quad (2.25)$$

- Setiap unit tersembunyi-j ($j = 1, 2, \dots, p$) memperbaiki bobot-i ($i = 1, 2, \dots, n$) dan biasnya;

$$v_{ij} \text{baru} = v_{ij} \text{lama} + \Delta v_{ij} \quad (2.26)$$

$$b 1_j \text{baru} = b 1_j \text{lama} + \Delta b 1_j \quad (2.27)$$

- vii. Hitung RMSE.

2.5.4 Root Mean Square Error (RMSE)

RMSE adalah salah satu metode pengukuran tingkat keakuratan suatu model peramalan yang didasarkan pada metode pengukuran *Mean Square Error* (MSE). Nilai RMSE diperoleh dari akar kuadrat dari nilai MSE (Hyndman dan Koehler, 2006).

RMSE lebih intuitif dibandingkan dengan MSE karena RMSE memiliki skala pengukuran yang sama dengan data yang sedang dievaluasi. Contohnya, eror baru dua kali nilai RMSE sebelumnya memiliki arti jika model melakukan kesalahan dua kali lebih besar dari sebelumnya. Namun eror baru dua kali nilai MSE sebelumnya tidak berarti demikian (Hyndman dan Koehler, 2006).

Rumus RMSE ditulis sebagai berikut.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - t_i)^2} \quad (2.28)$$

Keterangan: x_i = data hasil prediksi, t_i = data target yang diinginkan.

2.6 Normalisasi Data

Normalisasi data merupakan suatu proses mengubah nilai data ke dalam nilai yang berentang antara 0 dan 1. Normalisasi data dilakukan agar tidak ada variabel yang mendominasi dalam proses klasifikasi (Atthina dan Iswari, 2014).

rumus normalisasi data dapat dituliskan sebagai berikut.

$$y = \frac{x - D_{min}}{D_{max} - D_{min}} \quad (2.29)$$

Keterangan:

y = nilai data sesudah normalisasi,

x = nilai data sebelum normalisasi,

D_{max} = nilai maksimum dari seluruh data asli,

D_{min} = nilai minimum dari seluruh data asli.

2.7 Pengujian Akurasi

Dalam pengenalan pola, istilah pengujian akurasi banyak digunakan untuk mengukur kinerja dari suatu sistem. Pengujian akurasi adalah proses menguji tingkat kedekatan antara nilai prediksi dengan nilai aktual (Raharjo, 2011).

Rumus untuk mendapat nilai akurasi dapat dituliskan sebagai berikut.

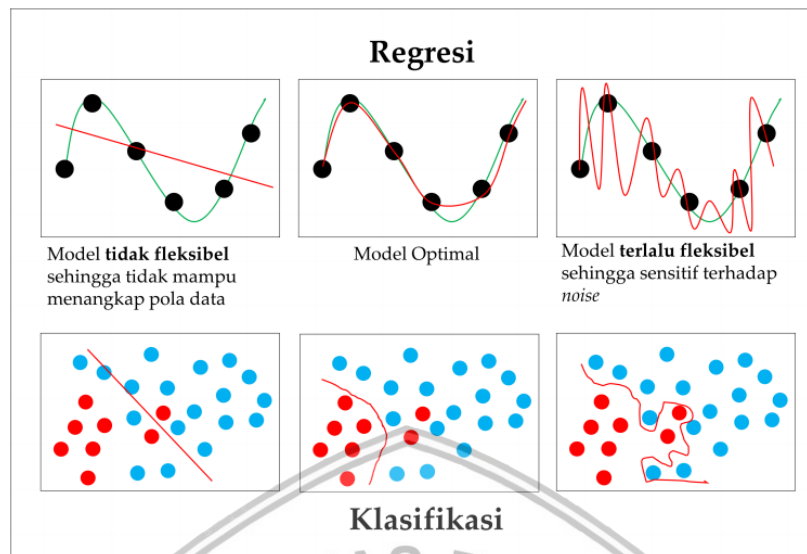
$$akurasi = \frac{\text{Jumlah Data Prediksi sesuai Target}}{\text{Jumlah Seluruh Data}} \times 100\% \quad (2.30)$$

2.8 Overfitting dan Underfitting

Dalam algoritme *machine learning* tidak selalu menghasilkan model yang mampu mengklasifikasikan data secara optimal (Putra, 2018), maksudnya nilai akurasi yang didapat dari evaluasi data latih tidak sebanding dengan nilai akurasi dari evaluasi data uji. Bisa saja nilai akurasi evaluasi data latih tinggi tetapi nilai akurasi evaluasi data ujinya rendah atau nilai akurasi dalam evaluasi data latih dan evaluasi data uji sama-sama rendah. Hal tersebut menunjukkan model yang dibangun mengalami kondisi *overfitting* atau *underfitting*.

Overfitting adalah keadaan ketika kinerja model memiliki nilai akurasi baik untuk *training* namun buruk untuk *testing*. Perhatikan Gambar 2.5 *overfitting* terjadi ketika model terlalu fleksibel (memiliki kemampuan yang terlalu tinggi untuk mencocokkan diri terhadap data latih) (Putra, 2018).

Underfitting adalah keadaan ketika kinerja model memiliki nilai akurasi yang buruk baik dalam *training* maupun *testing*. Lihat Gambar 2.5 *underfitting* terjadi akibat model yang tidak terlalu fleksibel (memiliki kemampuan yang rendah untuk menyesuaikan diri terhadap data latih) (Putra, 2018).



Gambar 2.5 Kondisi model *underfitting*, optimal, *overfitting*

Sumber: Putra (2018)

2.9 K-Fold Cross Validation

Cross-Validation merupakan metode evaluasi yang digunakan untuk mengetahui kinerja metode pembelajaran dalam *machine learning* dengan membagi *dataset* menjadi dua segmen/bagian (Refaeilzadeh, Tang dan Liu, 2008). Satu segmen digunakan untuk melatih model yang dibangun dan segmen yang lainnya digunakan untuk memvalidasi/mengevaluasi model dimana data dalam *dataset* harus di-*crossover* sehingga setiap data dalam *dataset* memiliki kesempatan untuk divalidasi. Dasar bentuk dari *cross-validation* adalah *k-fold cross validation*.

Dalam *k-fold cross validation*, *dataset* dibagi menjadi *k fold*/partisi/bagian dengan jumlah data yang hampir sama. Selanjutnya untuk setiap *fold-k* akan digunakan sebagai data uji/validasi (Refaeilzadeh, Tang dan Liu, 2008). Misalkan terdapat 100 data, dari data tersebut dibuatlah *4-fold cross validation* sehingga akan terbentuk empat *fold* yang masing-masing berjumlah 25 data. Selanjutnya dari *fold-1* akan menjadi data uji/validasi dan *fold-2* sampai 4 akan menjadi data latih. Selanjutnya *fold-2* menjadi data uji, *fold-1*, 3, 4 menjadi data latih, dan seterusnya sampai semua *fold* memiliki kesempatan menjadi data uji/validasi.

Tujuan dilakukan *k-fold cross validation* (Refaeilzadeh, Tang dan Liu, 2008):

- Untuk mengukur kemampuan suatu algoritme;
- Untuk membandingkan kinerja dua atau lebih algoritme sehingga dapat diketahui algoritme terbaik.

BAB 3 METODOLOGI

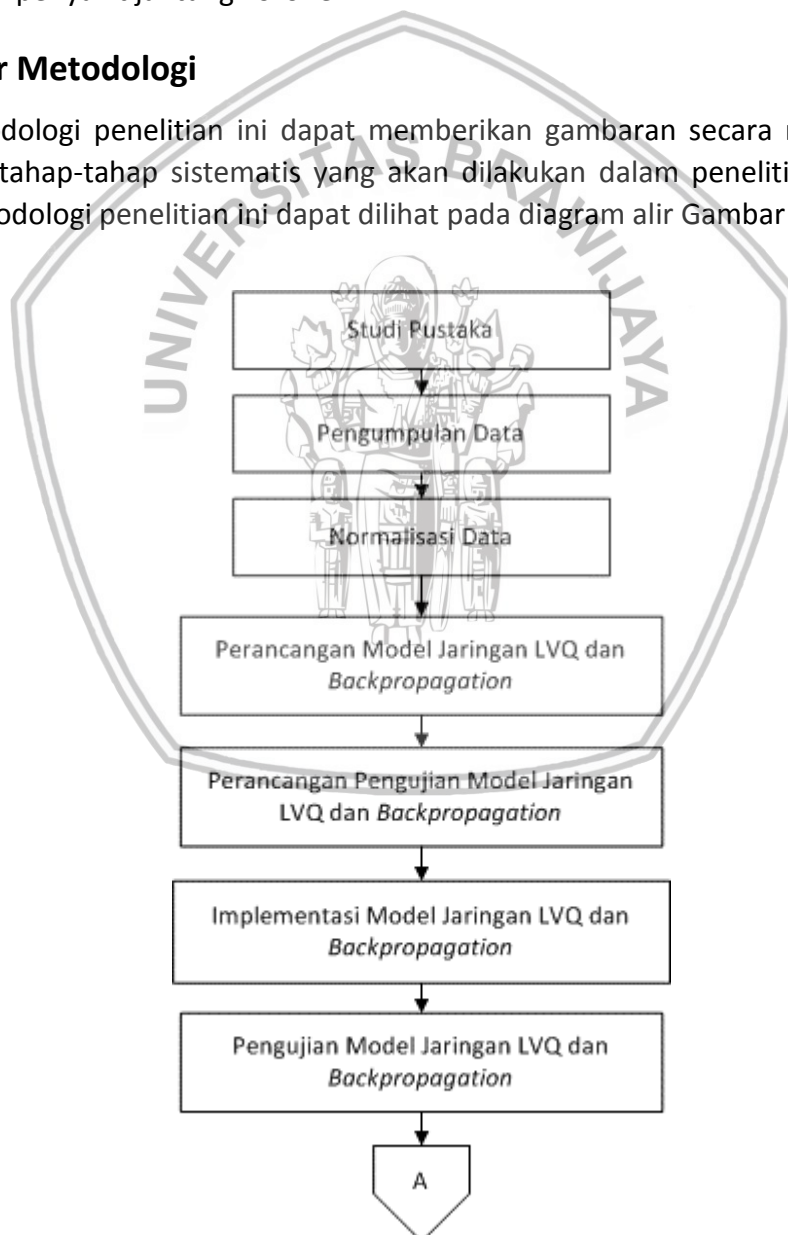
3.1 Tipe Penelitian

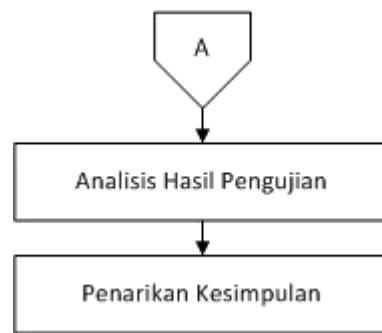
Berdasarkan produk/artefak utama yang dihasilkan dari proses penelitian skripsi ini yakni hasil analisis maka tipe penelitian skripsi ini termasuk dalam tipe penelitian non-implementatif analitik.

Penelitian ini berfokus pada penerapan metode komputasi cerdas untuk melakukan penyelesaian suatu permasalahan. Penelitian ini bertujuan untuk membandingkan metode LVQ dengan *Backpropagation* dalam permasalahan klasifikasi penyakit jantung koroner.

3.2 Alur Metodologi

Metodologi penelitian ini dapat memberikan gambaran secara menyeluruh tentang tahap-tahap sistematis yang akan dilakukan dalam penelitian. Adapun alur metodologi penelitian ini dapat dilihat pada diagram alir Gambar 3.1.





Gambar 3.1 Diagram alir metodologi

3.3 Studi Pustaka

Tahap ini adalah proses pengumpulan bahan, informasi, keterangan dan teori yang berasal dari buku, jurnal, dan website yang terkait dengan penelitian ini. Adapun teori-teori pendukung tersebut adalah:

1. Faktor-faktor penyakit jantung koroner;
2. Metode LVQ dan *Backpropagation*;
3. Pemrograman menggunakan bahasa pemrograman Java dengan IDE NetBeans.

3.4 Pengumpulan Data

Dalam penelitian ini dibutuhkan data-data masukkan untuk melakukan pengujian model yang dibangun. Pengumpulan data yang dilakukan adalah dengan cara mengambil data sekunder dari penelitian yang pernah dilakukan. Pengambilan data dari penelitian sebelumnya dilakukan agar mempermudah peneliti dalam melakukan penelitian.

Data yang dikumpulkan memiliki 10 fitur/variabel faktor penyakit jantung koroner, yakni: jenis kelamin, usia, denyut nadi, tekanan darah sistol, kolesterol, gula darah, trigliserida, nyeri dada, sesak nafas, dan batuk. Data pasien penyakit jantung koroner didapatkan dari penelitian Wardhani (2014).

3.5 Normalisasi Data

Dari data yang diperoleh, data akan dinormalisasi agar tidak ada fitur yang mendominasi dalam pengujian model. Data ternormalisasi akan dibagi menjadi data latih dan data uji untuk model.

3.6 Perancangan Model Jaringan LVQ dan *Backpropagation*

Tahap ini berfokus untuk merancang model jaringan LVQ dan *backpropagation*. Tahap-tahap yang akan dilakukan:

1. Merancang jaringan LVQ dan *Backpropagation*;
2. Merancang diagram alir metode LVQ dan *Backpropagation*;
3. Perhitungan manual metode LVQ dan *Backpropagation*.

3.7 Perancangan Uji Coba Model Jaringan LVQ dan *Backpropagation*

Selanjutnya untuk mempermudah proses pengujian akan dirancang rancangan Uji Coba Model Jaringan LVQ dan *Backpropagation*.

3.8 Implementasi Model Jaringan LVQ dan *Backpropagation*

Pada tahap ini seluruh hasil perancangan akan diimplementasikan ke dalam kode-kode bahasa pemrograman. Model Jaringan LVQ dan *Backpropagation* akan diimplementasikan dengan menggunakan bahasa pemrograman JAVA. Untuk mempermudah implementasi digunakan bantuan aplikasi editor IDE Netbeans dan berbagai *library*-nya.

3.9 Pengujian Model Jaringan LVQ dan *Backpropagation*

Tahap pengujian ini merupakan realisasi dari tahap perancangan uji coba model jaringan LVQ dan *backpropagation*. Fokus pengujian yang dilakukan adalah pengujian akurasi. Pengujian akurasi dilakukan dengan membandingkan hasil klasifikasi sistem dengan data asli.

3.10 Analisis Hasil Pengujian

Setelah pengujian dilakukan, dilakukan tahap analisis. Tahap analisis bertujuan untuk mengetahui hasil dari pengujian aplikasi sehingga dapat menarik pokok-pokok penting dari penelitian yang dilakukan yaitu perbandingan tingkat keakuratan antara model jaringan LVQ dan *Backpropagation* dalam mengklasifikasikan penyakit jantung koroner dan masalah yang dihadapi dalam penelitian ini.

3.11 Penarikan Kesimpulan

Penarikan kesimpulan diambil dari hasil analisis dan pengujian terhadap model yang dikembangkan dengan tujuan untuk mengetahui apakah permasalahan yang telah dirumuskan dapat terselesaikan. Dalam tahap ini juga disusun saran yang bertujuan untuk memperbaiki kesalahan dan menyempurnakan penelitian ke tahap selanjutnya.

BAB 4 PERANCANGAN

Bab ini menjelaskan mengenai formulasi permasalahan, perancangan model jaringan LVQ dan *Backpropagation*, diagram alir metode LVQ dan *Backpropagation*, perhitungan manual model jaringan LVQ dan *Backpropagation*, dan perancangan skenario uji coba.

4.1 Formulasi Permasalahan

Model jaringan LVQ dan *Backpropagation* yang dibangun digunakan untuk melakukan klasifikasi mengenai penyakit jantung koroner (PJK). Terdapat empat klasifikasi mengenai penyakit jantung koroner, yakni: normal, PJK tipe 1 (*Asimtomatik*), PJK tipe 2 (*Angina Pectoris*), dan PJK tipe 3 (*Infark Miokard Akut*). Data-data PJK akan disimpan dalam *file* berformat *.xls*. Data-data tersebut akan dibagi menjadi data latih dan data uji.

Hasil uji model jaringan LVQ dan *Backpropagation* selanjutnya akan dibandingkan tingkat akurasi. Sehingga didapatkan kesimpulan mengenai model jaringan mana yang cocok dalam permasalahan klasifikasi PJK.

Adapun data mengenai PJK yang digunakan memiliki 10 fitur. Sampel data ini ditunjukkan pada tabel 4.1.

4.2 Perancangan Model Jaringan

Perancangan model jaringan merupakan gambaran mengenai arsitektur jaringan saraf tiruan yang digunakan dalam penemitan ini.

4.2.1 LVQ

Gambar 4.1 menunjukkan arsitektur jaringan LVQ yang akan digunakan dalam penelitian ini. Dalam Gambar 4.1 terdapat dua *layer* yakni: *input layer* dan *ouput layer*.

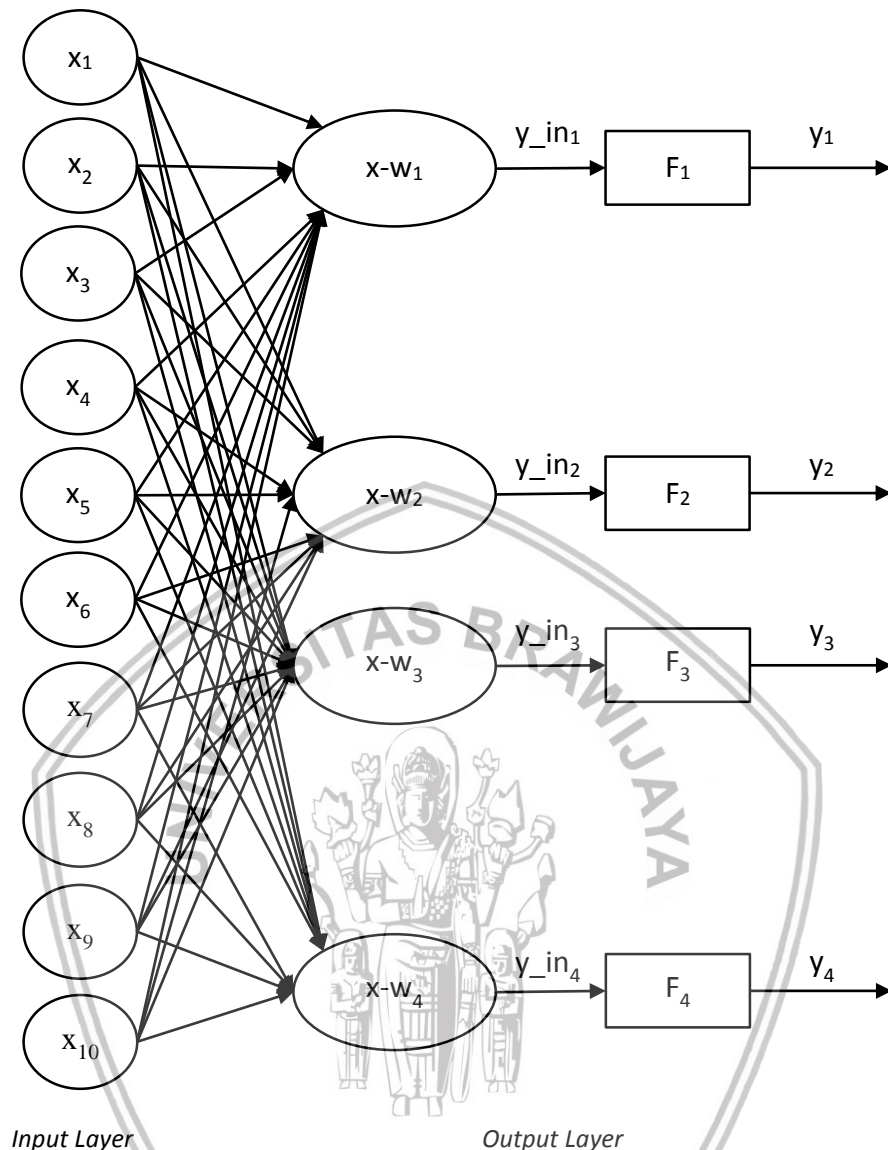
Input layer merupakan *layer* yang merepresentasikan banyak variabel yang digunakan. Dalam penelitian ini terdapat sepuluh variabel yang digunakan sebagai masukan, yakni: jenis kelamin, usia, denyut nadi, tekanan darah sistol, kolesterol, gula darah, trigliserida, nyeri dada, sesak nafas, dan batuk. Sehingga di dalam *layer input* terdapat sepuluh neuron atau *node*.

Tabel 4.1 Sampel data pasien penyakit jantung koroner

| Pasien | Jenis Kelamin | Usia | Denyut Nadi | Tekanan Darah Sistolic | Kolesterol | Gula Darah | Trigliserida | Nyeri Dada | Sesak Nafas | Batuk | Diagnosa |
|--------|---------------|------|-------------|------------------------|------------|------------|--------------|------------|-------------|-------|---------------|
| 1 | 0 | 46 | 120 | 200 | 151 | 381 | 480 | 1 | 1 | 1 | 3(PJK tipe 3) |
| 2 | 0 | 78 | 95 | 144 | 232 | 343 | 290 | 1 | 1 | 0 | 3(PJK tipe 3) |
| 3 | 0 | 59 | 90 | 100 | 96 | 456 | 330 | 0 | 1 | 1 | 3(PJK tipe 3) |
| 4 | 0 | 74 | 80 | 199 | 132 | 156 | 98 | 1 | 1 | 0 | 2(PJK tipe 2) |
| 5 | 0 | 64 | 90 | 190 | 162 | 262 | 60 | 0 | 1 | 1 | 2(PJK tipe 2) |
| 6 | 1 | 88 | 91 | 220 | 130 | 254 | 89 | 0 | 0 | 0 | 2(PJK tipe 2) |
| 7 | 1 | 70 | 80 | 120 | 144 | 550 | 190 | 0 | 0 | 1 | 1(PJK tipe 1) |
| 8 | 0 | 71 | 70 | 140 | 256 | 145 | 200 | 1 | 1 | 1 | 1(PJK tipe 1) |
| 9 | 1 | 60 | 83 | 140 | 133 | 154 | 97 | 1 | 1 | 0 | 1(PJK tipe 1) |
| 10 | 1 | 44 | 70 | 108 | 191 | 185 | 130 | 1 | 1 | 0 | 0(normal) |
| 11 | 1 | 42 | 74 | 92 | 189 | 102 | 95 | 0 | 1 | 0 | 0(normal) |
| 12 | 0 | 58 | 98 | 92 | 195 | 169 | 102 | 0 | 0 | 1 | 0(normal) |

Keterangan:

0 = laki-laki/tidak; 1 = perempuan/ya

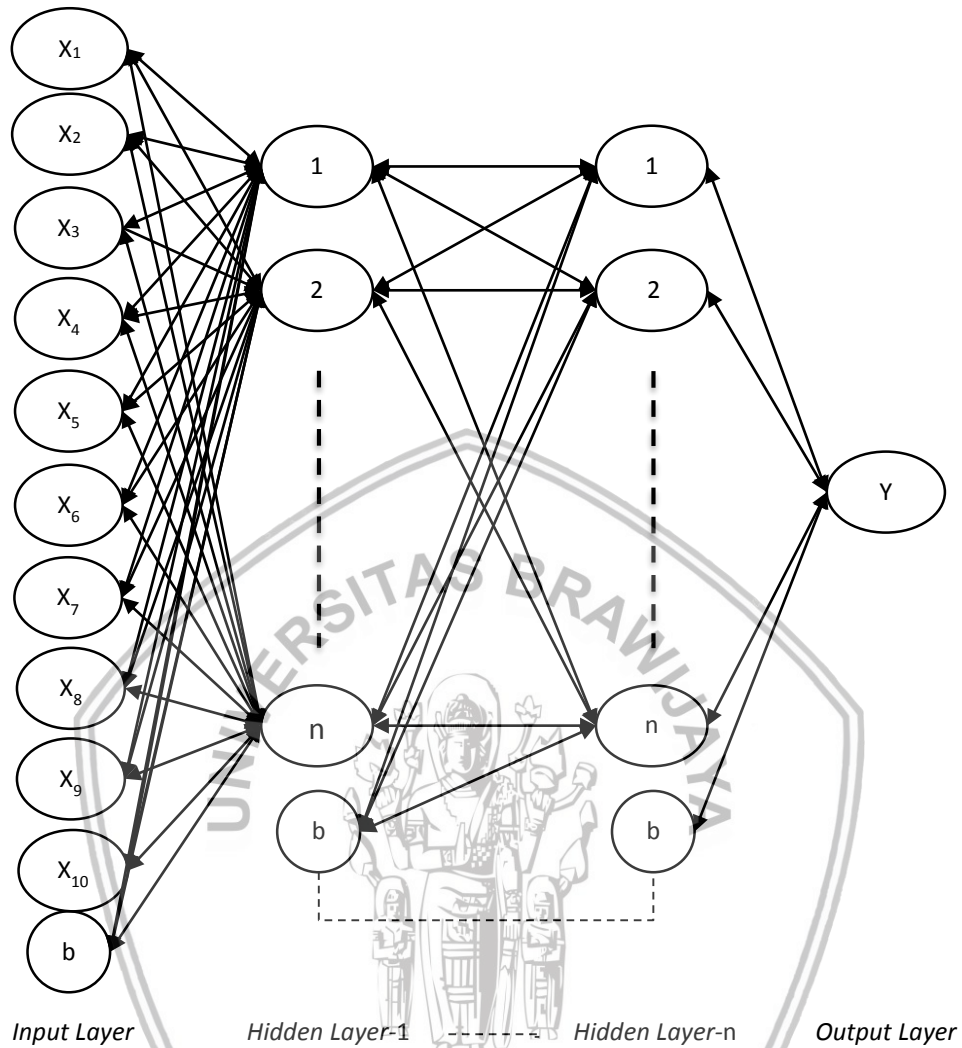


Gambar 4.1 Model jaringan LVQ

Output layer adalah *layer* yang mewakili banyaknya kelas yang terbentuk. Dalam penelitian ini data masukan akan dikelompokkan ke dalam empat kelas yang berbeda, yakni: normal, PJK tipe 1 (*Asimtomatik*), PJK tipe 2 (*Angina Pectoris*), dan PJK tipe 3 (*Infark Miokard Akut*). Sehingga dalam Gambar 4.1 terdapat empat neuron atau *node* dalam *layer output*.

Dalam *layer output* akan berlangsung proses pencarian jarak antara data masukan dengan ke-empat kelas tersebut. Jarak terdekat terhadap suatu kelas, akan menjadi kelas dari data masukan. Untuk lebih jelas lihat Diagram alir metode LVQ dan manualisasinya.

4.2.2 Backpropagation



Gambar 4.2 Model jaringan Backpropagation

Gambar 4.2 merupakan rancangan arsitektur jaringan *Backpropagation*. Dalam Gambar 4.2 terdapat beberapa *layer* yang di dalamnya terdapat banyak neuron. *Layer-layer* tersebut adalah *input layer*, *hidden layer*, dan *output layer*.

Input layer representasi dari banyaknya variabel data dan sebuah neuron bias, yakni neuron yang nilainya selalu 1. Dalam penelitian ini terdapat sepuluh variabel yang digunakan sebagai masukan, yakni: jenis kelamin, usia, denyut nadi, tekanan darah sistol, kolesterol, gula darah, trigliserida, nyeri dada, sesak nafas, dan batuk. Sehingga di dalam *layer input* terdapat sepuluh neuron atau *node*.

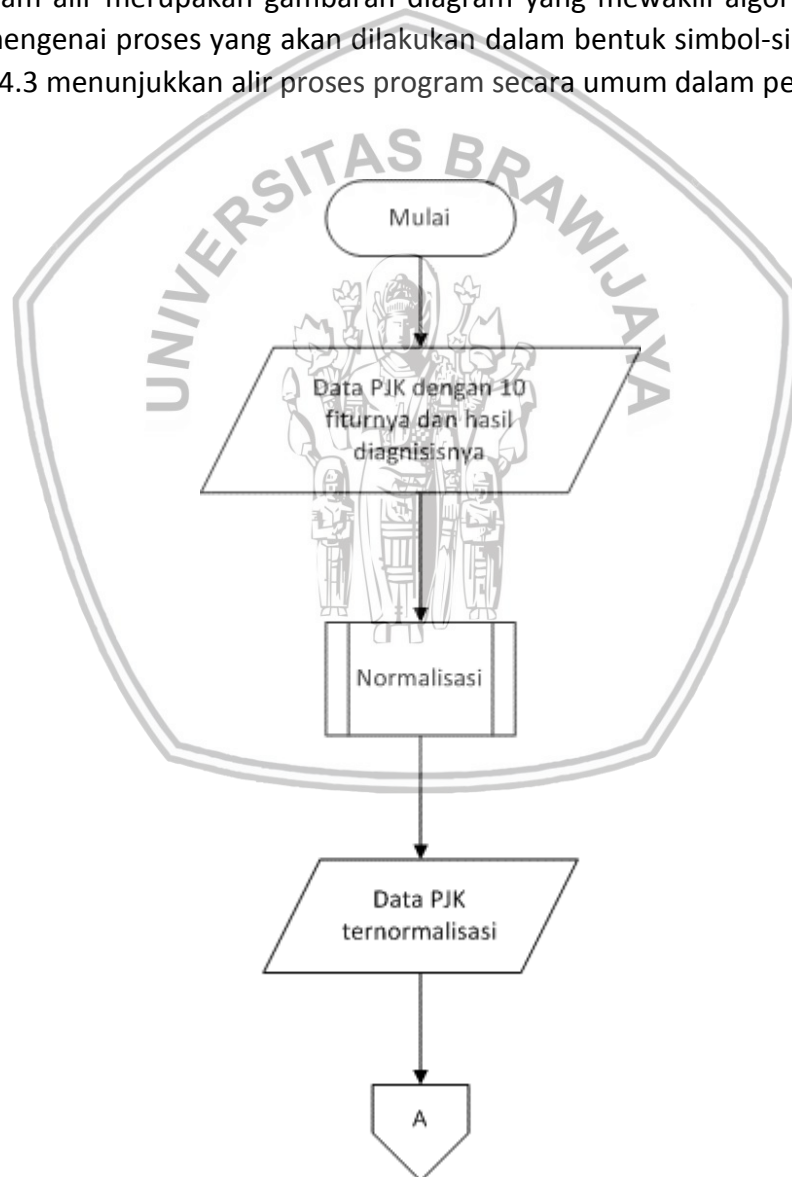
Hidden layer adalah *layer* yang bersifat dinamis, artinya banyak *layer* dan neuron dalam *layer* ini dapat diatur sesuai kebutuhan penelitian. Akan tetapi semakin banyak jumlah *layer* ini proses *Backpropagation* akan semakin lama.

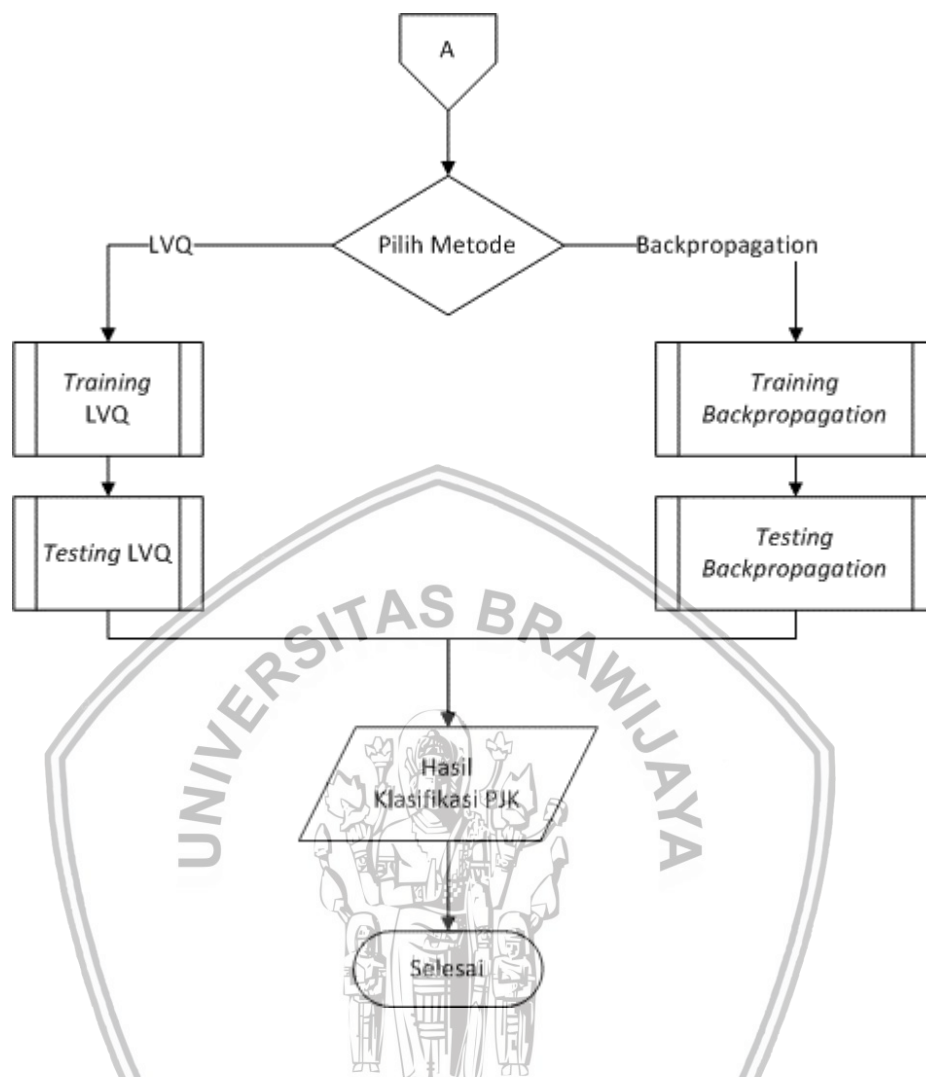
Output layer merupakan *layer* akhir hasil yang menunjukkan hasil prediksi atau klasifikasi. Jumlah neuron dalam *layer* ini disesuaikan dengan konteks permasalahan yang dihadapi. Karena dalam penelitian ini terdapat empat kelas, satu neuron dirasa sudah cukup digunakan untuk proses prediksi atau klasifikasi.

Dalam Gambar 4.2 setiap neuron dihubungkan dengan garis panah ganda. Hal itu dimaksudkan untuk menggambarkan proses pelatihan *backpropagation*, yang terdapat dua langkah yakni: langkah maju lalu langkah mundur. Setiap garis tersebut merepresentasikan bobot antar neuron.

4.3 Diagram Alir

Diagram alir merupakan gambaran diagram yang mewakili algoritme secara runtut mengenai proses yang akan dilakukan dalam bentuk simbol-simbol grafis. Gambar 4.3 menunjukkan alir proses program secara umum dalam penelitian ini.





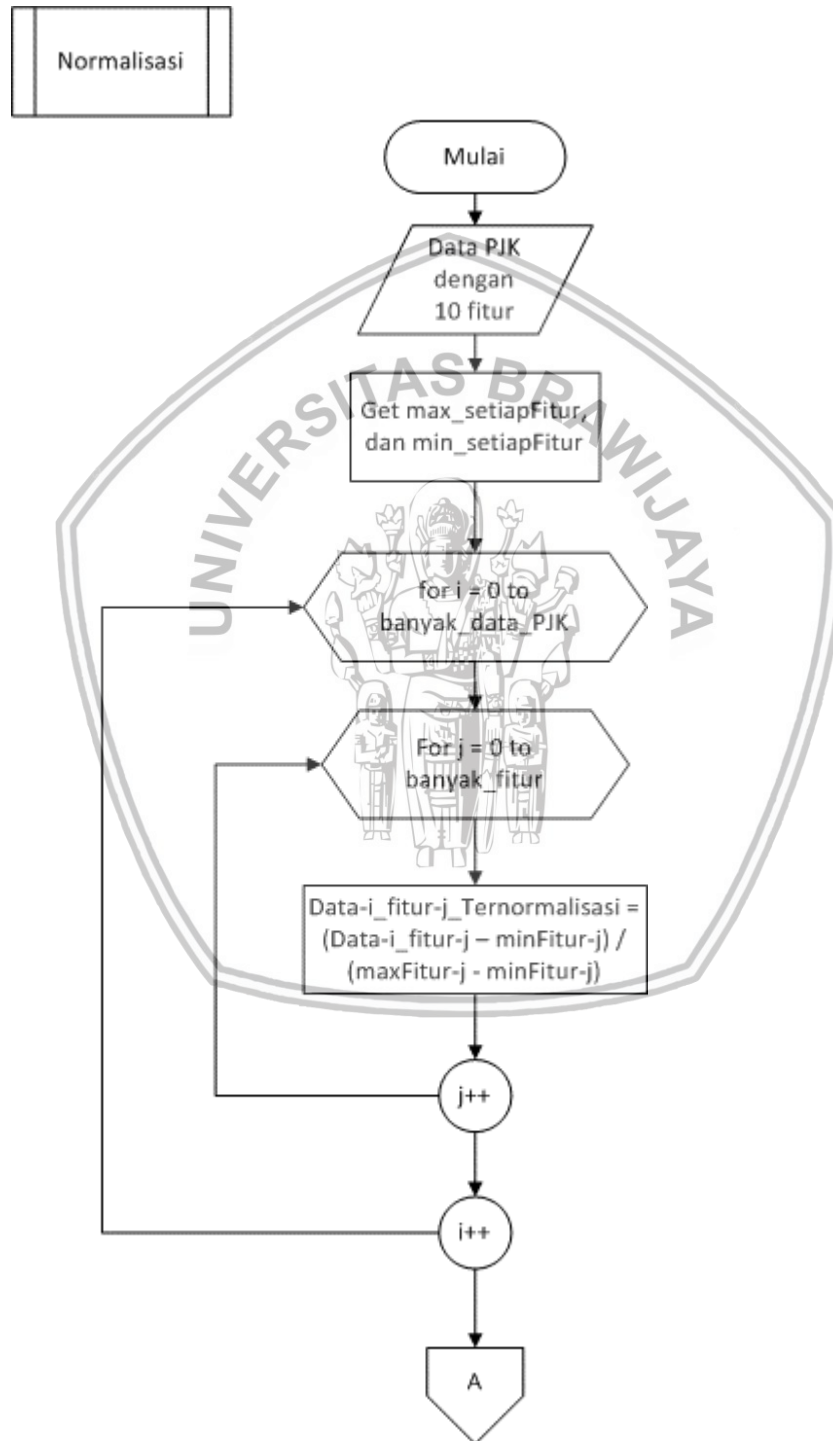
Gambar 4.3 Diagram alir program

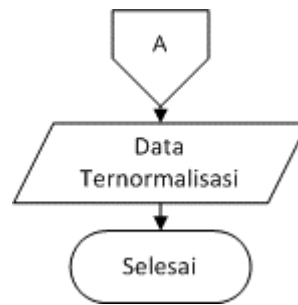
Penjelasan langkah-langkah proses dari Gambar 4.3 adalah sebagai berikut.

1. Masukkan data PJK (Penyakit Jantung Koroner) yang akan digunakan,
2. Lakukan proses normalisasi terhadap data masukkan, dengan Persamaan 2.29
3. Akan didapatkan data PJK yang sudah ternormalisasi yang akan dibagi menjadi data latih dan data uji.
4. Gunakan data latih untuk proses *training* untuk mendapat parameter yang dibutuhkan oleh LVQ atau *Backpropagation*:
 - a. Proses *training* dengan LVQ,
 - b. Proses *training* dengan *backpropagation*.
5. Lakukan proses *testing* dengan data uji untuk mendapatkan hasil klasifikasi atau prediksi mengenai PJK,
 - a. Proses *testing* dengan LVQ,
 - b. Proses *testing* dengan *backpropagation*.
6. Tampil Hasil Klasifikasi PJK

4.3.1 Normalisasi

Proses normalisasi dilakukan agar tidak ada fitur dalam data yang mendominasi dalam proses *training* dan *testing* model. Diagram alir dari proses normalisasi ditunjukkan pada Gambar 4.4.





Gambar 4.4 Diagram alir proses normalisasi

Berdasarkan Gambar 4.4, langkah-langkah proses normalisasi adalah sebagai berikut.

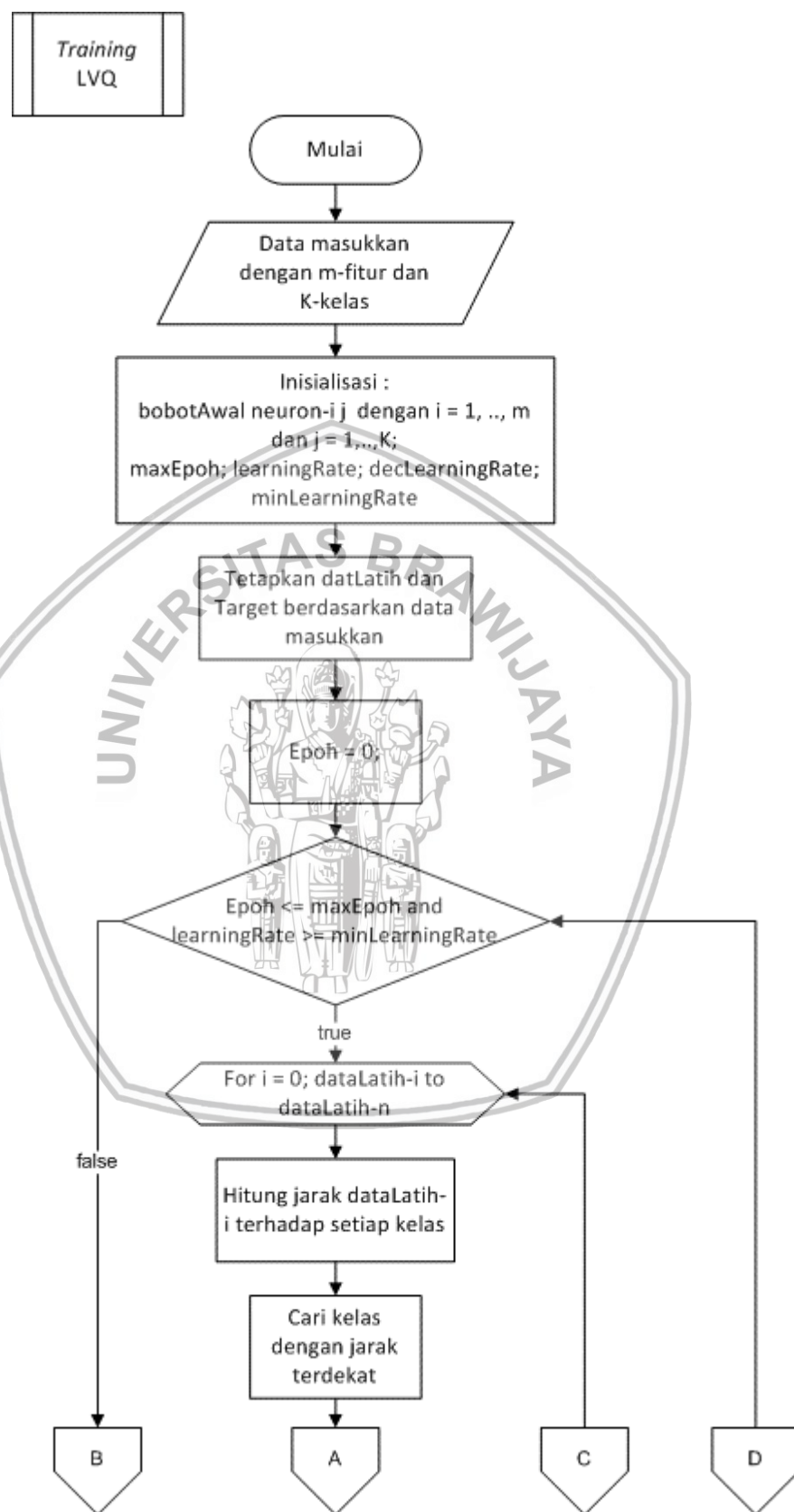
1. Masukkan data PJK yang akan dinormalisasi,
2. Dapatkan nilai maksimum dan nilai minimum setiap fitur/variabel dalam data,
3. Hitung normalisasi setiap data dalam data masukkan dengan Persamaan 2.29,
4. Didapatkan data yang sudah ternormalisasi.

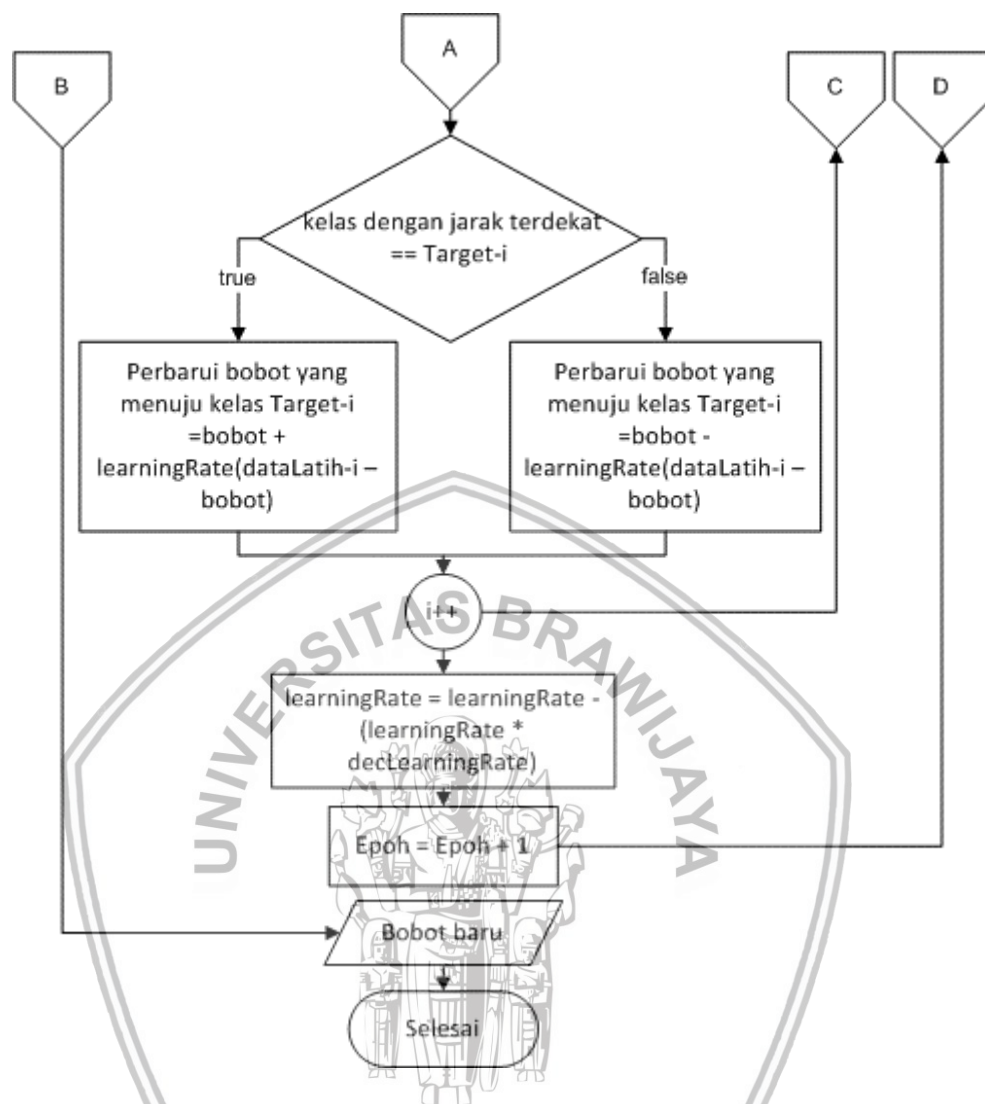
4.3.2 Metode LVQ

- **Training**

Langkah-langkah proses *training* dalam LVQ dapat dilihat dalam Gambar 4.5. Langkah-langkah tersebut dapat dijelaskan sebagai berikut.

1. Buat dua buah larik/array untuk menampung nilai data latih yang berisi fitur dan data target yang berisi target/kelas,
2. Tetapkan kondisi awal, $epoch = 0$,
3. Kerjakan jika: ($epoch \leq \text{maksimum } Epoch$) dan ($\text{learning rate} \geq \text{minimum learning rate}$),
 - a. Kerjakakan data latih-i sebanyak data latih-n,
 - i. Hitung jarak data latih ke i terhadap masing-masing kelas dengan Persamaan 2.1,
 - ii. Cari kelas dengan jarak terdekat,
 - iii. Jika kelas dengan jarak terdekat = data target ke- i
 1. Maka perbaiki bobot dengan Persamaan 2.2
 2. Jika tidak perbaiki bobot dengan Persamaan 2.3





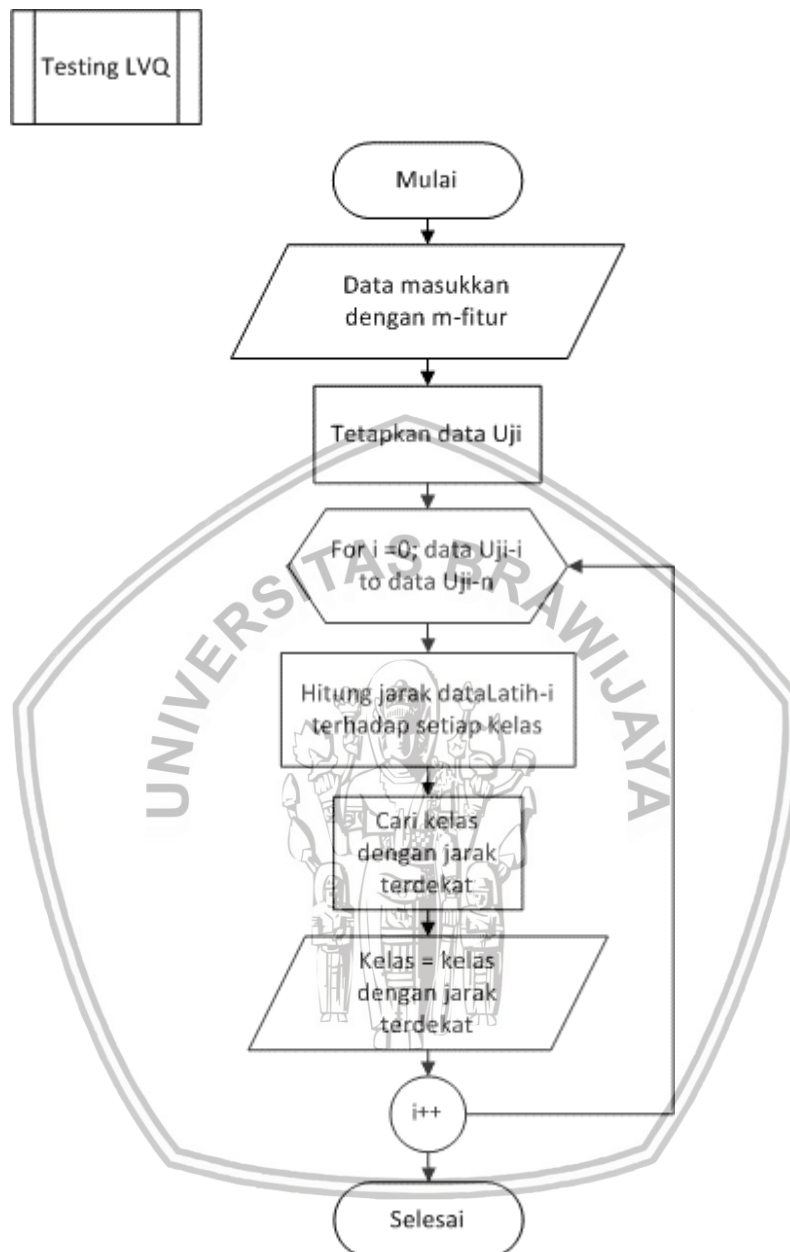
Gambar 4.5 Diagram alir proses *training* LVQ

- b. Perbarui nilai learning rate dengan Persamaan 2.4,
- c. Perbarui $epoch = epoch + 1$
4. Diperoleh bobot-bobot akhir.

• *Testing*

Langkah-langkah proses testing dalam LVQ digambarkan dalam diagram Gambar 4.6. Langkah-langkah tersebut dapat dijelaskan sebagai berikut.

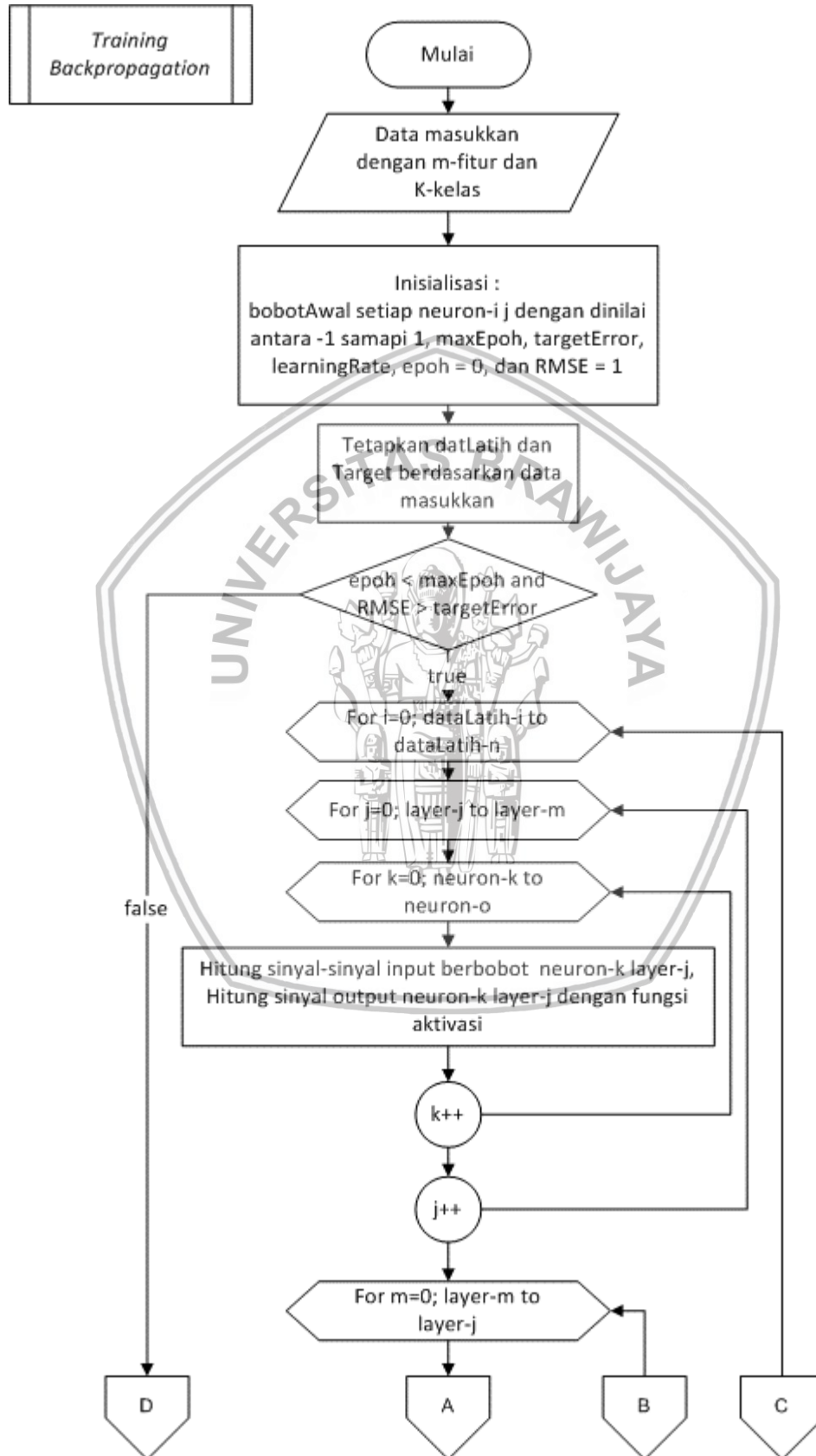
1. Masukkan data PJK yang digunakan untuk proses *testing*. Data PJK dalam penelitian ini memiliki sepuluh fitur,
2. Buat sebuah larik/array untuk menampung nilai data uji,

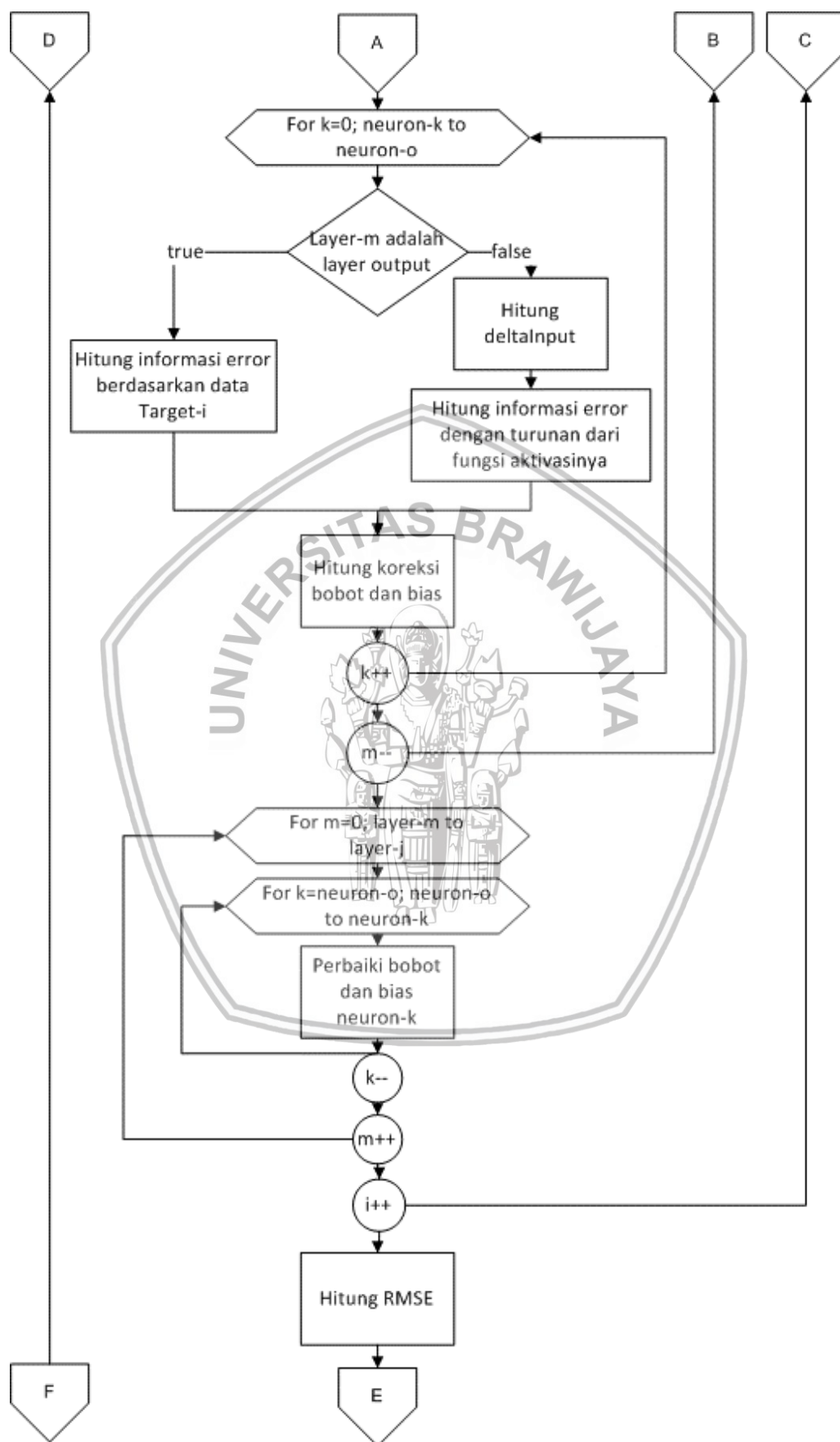


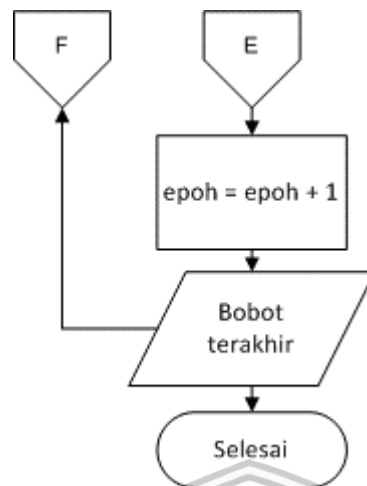
Gambar 4.6 Diagram alir proses *testing* LVQ

3. Kerjakan data Uji-i sebanyak data Uji-n
 - a. Hitung jarak data latih ke i terhadap masing-masing kelas dengan Persamaan 2.1,
 - b. Cari kelas dengan jarak terdekat,
 - c. Kelas dari data uji ke- i adalah kelas dengan jarak terdekat.

4.3.3 Metode *Backpropagation*







Gambar 4.7 Diagram alir proses *training Backpropagation*

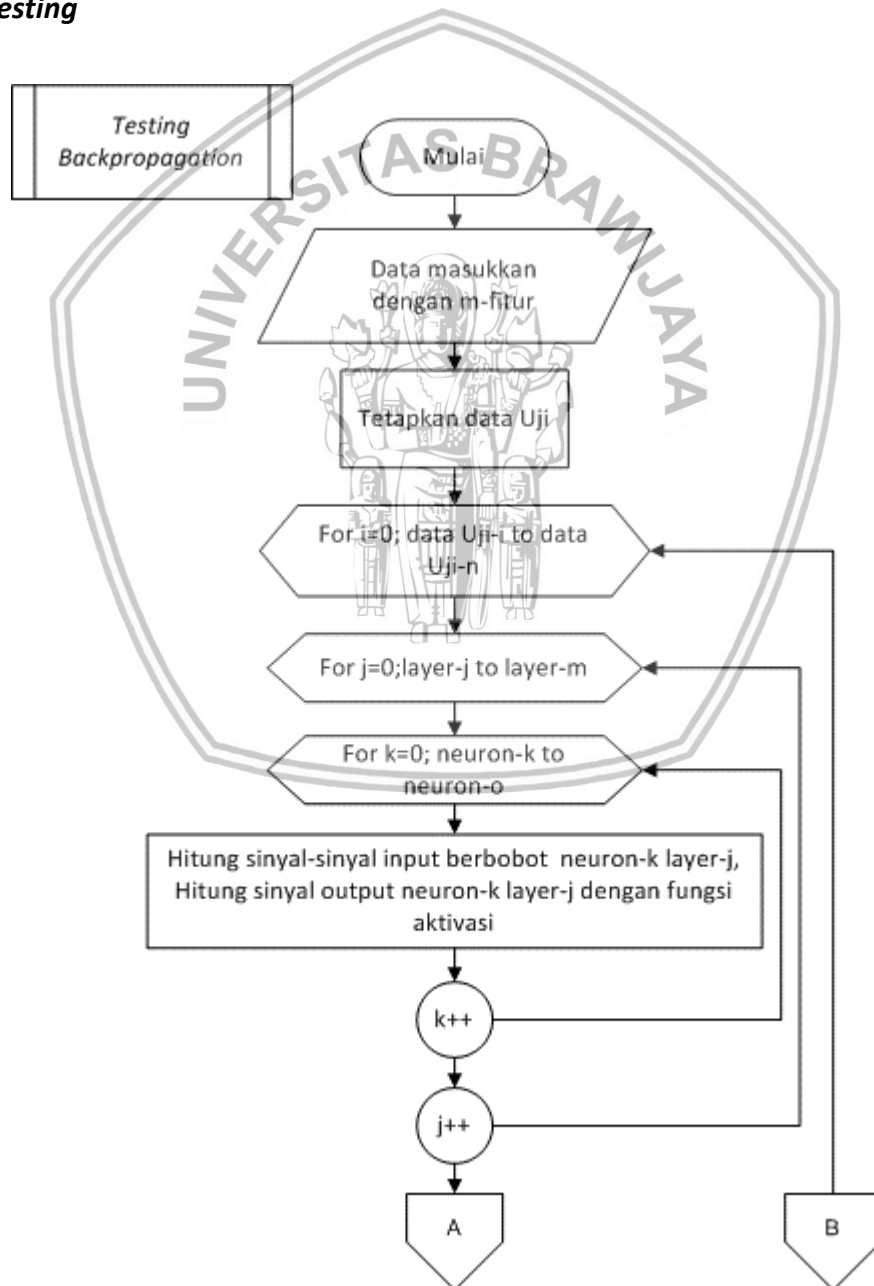
- **Training**

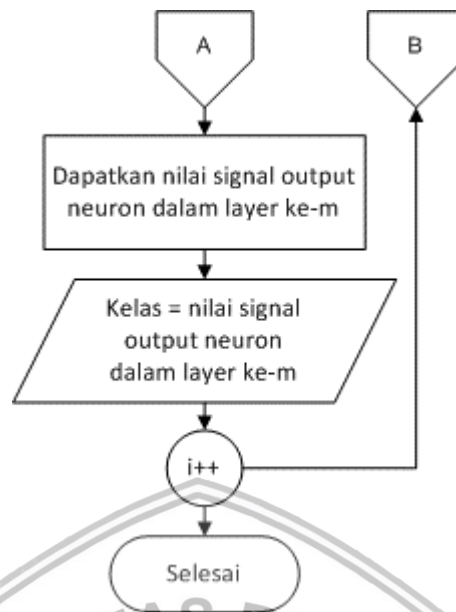
Gambar 4.7 menunjukkan langkah-langkah proses *training Backpropagation*. Langkah-langkahnya dijelaskan sebagai berikut.

1. Masukkan data PJK yang digunakan untuk proses *training*. Data PJK dalam penelitian ini memiliki sepuluh fitur dan empat kelas/target,
2. Melakukan inisialisasi terhadap bobot-bobot awal dengan nilai antara -1 sampai 1, *epoch* maksimum, target eror, *learning rate*, *epoch* = 0, dan RMSE = 1,
3. Buat dua buah larik/array untuk menampung nilai data latih yang berisi fitur dan data target yang berisi target/kelas,
4. Kerjakan: jika (*epoch* < *epoch* maksimum) dan (RMSE > target eror),
 - a. Ulangi sebanyak data Latih yang ada,
 - i. Ulangi sebanyak *layer* yang terbentuk,
 1. Untuk setiap neuron yang terdapat pada *layer* ke-*i*, hitung sinyal-sinyal berbobot dengan Persamaan 2.9, dan hitung sinyal *output*nya dengan Persamaan 2.10, sinyal-sinyal tersebut selanjutnya diteruskan ke setiap neuron yang berada dalam *layer* di atasnya/selanjutnya,
 - ii. Ulangi sebanyak *layer* yang terbentuk tetapi dimulai dengan *layer* yang paling akhir yakni *layer-m/layer output*
 1. Jika *layer* ke-*m* adalah *layer output*, maka hitung informasi eror berdasarkan Target ke-*i* sesuai Persamaan 2.13-2.15, gunakan informasi eror ini untuk perhitungan informasi eror dalam *layer* selanjutnya/di bawahnya. Hitung koreksi bobot dan bias dengan Persamaan 2.16 dan 2.17,

2. Jika tidak, maka hitung $\Delta Input$ setiap neuron dengan Persamaan 2.18 lalu hitung informasi eror dengan Persamaan 2.19 sampai 2.21. Hitung koreksi bobot dan bias dengan Persamaan 2.22 dan 2.23,
- iii. Ulangi sebanyak *layer* yang terbentuk
 1. Untuk setiap neuron, hitung perbaikan bobot dan bias dengan Persamaan 2.24 dan 2.25,
 - b. Hitung RMSE dengan Persamaan 2.28, perbarui nilai *epoch*; $epoch = epoch + 1$,
5. Diperoleh bobot-bobot akhir.

• **Testing**





Gambar 4.8 Diagram alir proses *testing Backpropagation*

Gambar 4.8 menunjukkan langkah-langkah proses *testing Backpropagation* yang dapat dijelaskan sebagai berikut.

1. Masukkan data PJK yang digunakan untuk proses *testing*. Data PJK dalam penelitian ini memiliki sepuluh fitur,
2. Buat sebuah larik/array untuk menampung nilai data uji,
3. Kerjakan data Uji-i sebanyak data Uji-n
 - a. Ulangi sebanyak *layer* yang terbentuk/sampai *layer-m*,
 - i. Untuk setiap neuron yang terdapat pada *layer* ke-i, hitung sinyal-sinyal berbobot dengan Persamaan 2.9, dan hitung sinyal *output*nya dengan Persamaan 2.10, sinyal-sinyal tersebut selanjutnya diteruskan ke setiap neuron yang berada dalam *layer* di atasnya/selanjutnya,
4. Nilai kelas diperoleh dari nilai sinyal *output* neuron dalam *layer-m*.

4.4 Perhitungan Manual

Perhitungan manual merupakan penerapan dari rancangan diagram alir yang digunakan untuk memvalidasi rancangan diagram alir.

4.4.1 Normalisasi Data

Dalam manualisasi ini, data yang digunakan berjumlah 12 data dan memiliki 10 fitur yakni, jenis kelamin, usia, denyut nadi, tekanan darah sistol, kolestrol, gula darah, trigliserida, nyeri dada, sesak nafas, dan batuk. Data dapat dilihat pada Tabel 4.1.

Langkah-langkah manualisasi dari normalisasi data adalah sebagai berikut.

1. Masukkan data PJK yang akan dinormalisasi, lihat Tabel 4.1
2. Dapatkan nilai maksimum dan nilai minimum setiap fitur/variabel dalam data, yang ditunjukkan Tabel 4.2

Tabel 4.2 Nilai maksimum dan minimum setiap fitur

| Fitur | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|-------|---|----|-----|-----|-----|-----|-----|---|---|----|
| Max | 1 | 88 | 120 | 220 | 256 | 550 | 480 | 1 | 1 | 1 |
| Min | 0 | 42 | 70 | 92 | 96 | 102 | 60 | 0 | 0 | 0 |

3. Hitung normalisasi setiap data dalam data masukkan dengan Persamaan 2.29, contoh.

$$\text{normalisasi fitur kedua} = \frac{x - D_{\min}}{D_{\max} - D_{\min}}$$

$$\text{normalisasi fitur kedua} = \frac{78 - 42}{88 - 42}$$

$$\text{normalisasi fitur kedua} = 0,782609$$

Tabel 4.3 Hasil normalisasi data

| Fitur Data | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|------------|---|----------|------|----------|---------|----------|----------|---|---|----|
| 1 | 0 | 0,782609 | 0,5 | 0,40625 | 0,85 | 0,537946 | 0,547619 | 1 | 1 | 1 |
| 2 | 0 | 0,369565 | 0,4 | 0,0625 | 0 | 0,790179 | 0,642857 | 1 | 1 | 0 |
| 3 | 0 | 0,695652 | 0,2 | 0,835938 | 0,225 | 0,120536 | 0,090476 | 0 | 1 | 1 |
| 4 | 0 | 1 | 0,42 | 1 | 0,2125 | 0,339286 | 0,069048 | 1 | 1 | 0 |
| 5 | 0 | 0,608696 | 0,2 | 0,21875 | 0,3 | 1 | 0,309524 | 0 | 1 | 1 |
| 6 | 1 | 0,630435 | 0 | 0,375 | 1 | 0,095982 | 0,333333 | 0 | 0 | 0 |
| 7 | 1 | 0,043478 | 0 | 0,125 | 0,59375 | 0,185268 | 0,166667 | 0 | 0 | 1 |
| 8 | 0 | 0 | 0,08 | 0 | 0,58125 | 0 | 0,083333 | 1 | 1 | 1 |
| 9 | 1 | 0,086957 | 1 | 0,84375 | 0,34375 | 0,622768 | 1 | 1 | 1 | 0 |
| 10 | 1 | 0,478261 | 0,4 | 0,765625 | 0,4125 | 0,357143 | 0 | 1 | 1 | 0 |
| 11 | 1 | 0,391304 | 0,26 | 0,375 | 0,23125 | 0,116071 | 0,088095 | 0 | 1 | 0 |
| 12 | 0 | 0,347826 | 0,56 | 0 | 0,61875 | 0,149554 | 0,1 | 0 | 0 | 1 |

4.4.2 Metode LVQ

Sebelum melakukan proses *training* dan *testing* metode LVQ dibentuk arsitektur jaringannya. Arsitektur jaringan yang terbentuk adalah 10 neuron dalam *layer input* dan 4 neuron dalam *layer output*, arsitektur jaringannya dapat dilihat pada Gambar 4.1. Sehingga dari arsitektur jaringan tersebut terbentuk 40 bobot.

Tabel 4.4 Bobot yang terbentuk

| $i \backslash J$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|------------------|----------|---|---|---|---|---|---|---|---|-----------|
| 1 | W_{11} | | | | | | | | | |
| 2 | | | | | | | | | | |
| 3 | | | | | | | | | | |
| 4 | | | | | | | | | | W_{410} |

4.4.2.1 Training LVQ

Langkah manualisasi *training* LVQ adalah sebagai berikut.

1. Masukkan data PJK yang digunakan untuk proses *training*. Data PJK dalam penelitian ini memiliki sepuluh fitur dan empat kelas/target,
2. Melakukan inisialisasi terhadap bobot-bobot awal dengan data masukkan berdasarkan target kelas, contohnya: bobot yang menuju *output layer* neuron kelas 1 akan diisi dengan nilai data masukkan dengan target 1, demikian seterusnya sampai semua bobot memiliki nilai awal. Lalu inisialisasikan maksimum *epoch*, *learning rate*, pengurangan *learning rate*, dan minimum *learning rate* yang diperbolehkan,

Tabel 4.4 Inisialisasi bobot

| $i \backslash J$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|------------------|---|----------|------|---------|---------|----------|----------|---|---|----|
| 1 | 0 | 0 | 0,08 | 0 | 0,58125 | 0 | 0,083333 | 1 | 1 | 1 |
| 2 | 0 | 0,608696 | 0,2 | 0,21875 | 0,3 | 1 | 0,309524 | 0 | 1 | 1 |
| 3 | 0 | 1 | 0,42 | 1 | 0,2125 | 0,339286 | 0,069048 | 1 | 1 | 0 |
| 4 | 0 | 0,782609 | 0,5 | 0,40625 | 0,85 | 0,537946 | 0,547619 | 1 | 1 | 1 |

3. Hitung jarak data latih ke $-i$ terhadap masing-masing kelas dengan Persamaan 2.1,

$$jarak = \sqrt{(x_1 - w_{11})^2 + (x_2 - w_{12})^2 + \dots + (x_{10} - w_{110})^2}$$

$$jarak = \sqrt{(0 - 0)^2 + (0,782609 - 0)^2 + \dots + (1 - 1)^2}$$

$$jarak = 1,237372$$

Tabel 4.5 Jarak data latih ke-1 terhadap setiap kelas

| | | |
|-----------------------|---|----------|
| Jarak data-1 ke kelas | 0 | 1,237372 |
| | 1 | 1,314566 |
| | 2 | 1,442602 |
| | 3 | 1,842137 |

4. Cari Jarak terdekat, Jika kelas dengan jarak terdekat = data target ke-i maka perbaiki bobot dengan Persamaan 2.2. Jika tidak perbaiki bobot dengan Persamaan 2.3,
5. Perbarui nilai parameter *learning rate* dengan Persamaan 2.4,

Tabel 4.6 Jarak data latih ke-1 terhadap setiap kelas

| | Prediksi | Target aktual |
|--------------------------|----------|---------------|
| Jarak terdekat ke kelas- | 0 | 3 |

Tabel 4.7 Perbaikan bobot

| i \ J | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|-------|---|----------|-------|----------|----------|----------|----------|---|---|----|
| 1 | 0 | -0,07826 | 0,038 | -0,04063 | 0,554375 | -0,05379 | 0,036905 | 1 | 1 | 1 |
| 2 | 0 | 0,608696 | 0,2 | 0,21875 | 0,3 | 1 | 0,309524 | 0 | 1 | 1 |
| 3 | 0 | 1 | 0,42 | 1 | 0,2125 | 0,339286 | 0,069048 | 1 | 1 | 0 |
| 4 | 0 | 0,782609 | 0,5 | 0,40625 | 0,85 | 0,537946 | 0,547619 | 1 | 1 | 1 |

6. Ulangi langkah 3 dan 4 sebanyak data latih dan banyak *epoch* serta sesuai aturan berhentinya, sehingga diperoleh bobot akhir hasil proses *training*.

Tabel 4.8 Bobot akhir

| i \ J | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|-------|----------|----------|----------|----------|----------|----------|----------|---|----------|----------|
| 1 | 0 | -0,0819 | 0,089828 | -0,0819 | 0,611451 | -0,02779 | 0,037512 | 1 | 1 | 1,0819 |
| 2 | 0,017939 | 0,704973 | 0,391218 | 0,164407 | 0,401757 | 1,115504 | 0,512757 | 0 | 0,982061 | 0,793675 |
| 3 | 0 | 1 | 0,8022 | 1 | 0,2125 | 0,339286 | 0,006214 | 1 | 1 | 0 |

| | | | | | | | | | | |
|---|---|--------------|--------------|--------------|--------------|--------------|--------------|---|---|-------------|
| 4 | 0 | 0,7112 31 | 0,8884 88 | 0,3468 47 | 0,7031 12 | 0,5815 35 | 0,8071 08 | 1 | 1 | 0,8271 9 |
|---|---|--------------|--------------|--------------|--------------|--------------|--------------|---|---|-------------|

4.4.2.2 Testing LVQ

Langkah manualisasi *testing* LVQ adalah sebagai berikut.

1. Masukkan data PJK yang digunakan untuk proses *testing*. Data PJK dalam penelitian ini memiliki sepuluh fitur,

Tabel 4.9 Sample data uji

| Fitur Data | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---------------|---|----------|------|----------|---------|----------|----------|---|---|----|
| 1 | 1 | 0,086957 | 1 | 0,84375 | 0,34375 | 0,622768 | 1 | 1 | 1 | 0 |
| 2 | 1 | 0,478261 | 0,4 | 0,765625 | 0,4125 | 0,357143 | 0 | 1 | 1 | 0 |
| 3 | 1 | 0,391304 | 0,26 | 0,375 | 0,23125 | 0,116071 | 0,088095 | 0 | 1 | 0 |
| 4 | 0 | 0,347826 | 0,56 | 0 | 0,61875 | 0,149554 | 0,1 | 0 | 0 | 1 |

2. Hitung jarak data latih ke-*i* terhadap masing-masing kelas dengan Persamaan 2.1, Cari kelas dengan jarak terdekat. Kelas dari data uji ke-*i* adalah kelas dengan jarak terdekat.

$$jarak = \sqrt{(x_1 - w_{11})^2 + (x_2 - w_{12})^2 + \dots + (x_n - w_{1n})^2}$$

$$jarak = \sqrt{(1 - 0)^2 + (0,086957 - 0,0819)^2 + \dots + (0 - 1,0819)^2}$$

$$jarak = 2,303373$$

Tabel 4.10 Hasil testing

| Data | Jarak Terhadap ke Target | | | | Jarak terkecil | Hasil Prediksi |
|------|--------------------------|----------|----------|----------|----------------|----------------|
| | 0 | 1 | 2 | 3 | | |
| 1 | 2,303373 | 2,07179 | 1,726959 | 1,581568 | 1,581568145 | 3 |
| 2 | 1,867606 | 1,961064 | 1,236635 | 1,714276 | 1,236634827 | 2 |
| 3 | 1,949347 | 1,721312 | 1,764085 | 2,034373 | 1,721312322 | 1 |
| 4 | 1,5667 | 1,530007 | 2,166524 | 1,756092 | 1,530007023 | 1 |

4.4.3 Metode Backpropagation

Arsitektur jaringan *backpropagation* yang akan dilakukan manualisasi adalah jaringan dengan 11 neuron (sudah termasuk 1 buah bias) dalam *layer input*, 3 neuron (sudah termasuk 1 buah bias) dalam *layer hidden*, dan 1 neuron dalam *layer output*. Fungsi aktivasi yang digunakan adalah *sigmoid* biner untuk *layer input* dan *layer hidden*, dan fungsi identitas yang dibulatkan untuk *layer output*.

Tabel 4.11 Bobot yang terbentuk

| LI -> LH | | | | | | | | | | | LH -> LO |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|------|-----|----------|
| w11 | w21 | w31 | w41 | w51 | w61 | w71 | w81 | w91 | w101 | b11 | v11 |
| | | | | | | | | | | | |
| w12 | w22 | w32 | w42 | w52 | w62 | w72 | w82 | w92 | w102 | b12 | v21 |
| | | | | | | | | | | | |
| | | | | | | | | | | | b21 |
| | | | | | | | | | | | |

4.4.3.1 Training Backpropagation

Langkah-langkah manualisasi proses *training Backpropagation* adalah sebagai berikut.

1. Masukkan data PJK yang digunakan untuk proses *training*. Data PJK dalam penelitian ini memiliki sepuluh fitur dan empat kelas/target,
2. Melakukan inialisasi terhadap bobot-bobot awal dengan nilai antara -1 sampai 1, *epoch* maksimum, target error, *learning rate*, *epoch* = 0, dan RMSE = 1,

Tabel 4.12 Inisialisasi bobot

| LI -> LH | | | | | | | | | | | LH -> LO |
|----------|------|------|------|------|------|------|------|------|------|------|----------|
| w11 | w21 | w31 | w41 | w51 | w61 | w71 | w81 | w91 | w101 | b11 | v11 |
| - | - | - | - | - | - | - | - | - | - | - | - |
| 0,30 | 0,16 | 0,25 | 0,25 | 0,01 | 0,27 | 0,08 | 0,48 | 0,19 | 0,47 | 0,35 | 0,31422 |
| 408 | 524 | 418 | 376 | 538 | 550 | 409 | 939 | 822 | 203 | 601 | 3375 |
| w12 | w22 | w32 | w42 | w52 | w62 | w72 | w82 | w92 | w102 | b12 | v21 |
| - | - | - | - | - | - | - | - | - | - | - | - |
| 0,07 | 0,08 | 0,07 | 0,46 | 0,06 | 0,37 | 0,17 | 0,30 | 0,23 | 0,44 | 0,41 | 0,39554 |
| 211 | 304 | 301 | 074 | 369 | 007 | 887 | 305 | 333 | 440 | 478 | 6915 |
| | | | | | | | | | | | b21 |
| | | | | | | | | | | | 0,20022 |
| | | | | | | | | | | | 335 |

3. Untuk setiap neuron yang terdapat pada *layer* ke-i, hitung sinyal-sinyal berbobot dengan Persamaan 2.9, dan hitung sinyal *output*nya dengan Persamaan 2.10, sinyal-sinyal tersebut selanjutnya diteruskan ke setiap neuron yang berada dalam *layer* diatasnya/selanjutnya, contoh:

$$z_{in_j} = b1_j + \sum_{i=1}^n x_i \times bobot_{ij}$$

$$z_{in_j} = 0,35601 + (0 \times -0,30408 + 0,7826209 \times 0,16524 + \dots + 1 \times 0,47203$$

$$z_{in_j} = 0,871365$$

$$z_j = f(z_{in_j})$$

$$z_j = \frac{1}{1 + e^{-z_{in_j}}}$$

$$z_j = \frac{1}{1 + e^{-0,871365}}$$

$$z_j = 0,70503$$

Tabel 4.13 Hasil signal pada data latih pertama

| Proses <i>Layer</i> tersembunyi | | | | | Proses <i>Layer Output</i> | | | | |
|---------------------------------|--------------|----------|---------------|----------|----------------------------|--------------|----------|---------------|---|
| Neuron 1 | <i>Input</i> | 0,871365 | <i>Output</i> | 0,70503 | Neuron 1 | <i>Input</i> | 0,255779 | <i>Output</i> | 0 |
| Neuron 2 | <i>Input</i> | -0,32432 | <i>Output</i> | 0,419624 | | | | | |

4. Jika *layer* ke-*m* adalah *layer output*, maka hitung informasi error berdasarkan Target ke-*i* sesuai Persamaan 2.13 sampai 2.15, gunakan informasi error ini untuk perhitungan informasi error dalam *layer* selanjutnya/di bawahnya. Hitung koreksi bobot dan bias dengan Persamaan 2.16 dan 2.17,

Jika tidak, maka hitung delta *input* setiap neuron dengan Persamaan 2.18 lalu hitung informasi error dengan Persamaan 2.19 sampai 2.21. Hitung koreksi bobot dan bias dengan Persamaan 2.22 dan 2.23,

$$\delta 2_k = (t_k - y_k) f'(y_{in_k})$$

$$\delta 2_k = (3 - 0) * 1; \text{turunan fungsi identitas adalah 1}$$

$$\delta 2_k = 3$$

$$\omega 2_{jk} = \delta 2_k z_j$$

$$\omega 2_{jk} = 3 * 0,70503$$

$$\beta 2_k = \delta 2_k$$

$$\beta 2_k = 3$$

$$\Delta w_{jk} = \alpha \omega 2_{jk}$$

$$\Delta w_{jk} = 0,1 * 3 * 0,70503$$

$$\Delta w_{jk} = 0,211509$$

$$\Delta b 2_k = \alpha \beta 2_k$$

$$\Delta b 2_k = 0,1 * 3$$

$$\Delta b 2_k = 0,3$$

Tabel 4.14 Hasil koreksi bobot data latih-1

| Proses <i>Layer output</i> | | | |
|----------------------------|-----------------|-----------------|-----------------|
| Delta | Δv_{11} | Δv_{21} | Δb_{21} |
| 3 | 0,211509 | 0,125887 | 0,3 |

Tabel 4,14 (lanjutan)

| Proses Layer tersembunyi | | | | | | | | | | | |
|--------------------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|------------------|-----------------|
| Delta-1 | Δw_{11} | Δw_{21} | Δw_{31} | Δw_{41} | Δw_{51} | Δw_{61} | Δw_{71} | Δw_{81} | Δw_{91} | Δw_{101} | Δb_{11} |
| 0,105662 | 0 | 0,008269 | 0,005283 | 0,004293 | 0,008981 | 0,005684 | 0,005786 | 0,010566 | 0,010566 | 0,010566 | 0,010566 |
| Delta-2 | Δw_{12} | Δw_{22} | Δw_{32} | Δw_{42} | Δw_{52} | Δw_{62} | Δw_{72} | Δw_{82} | Δw_{92} | Δw_{102} | Δb_{12} |
| 0,50966 | 0 | 0,039886 | 0,025483 | 0,020705 | 0,043321 | 0,027417 | 0,02791 | 0,050966 | 0,050966 | 0,050966 | 0,050966 |

5. Untuk setiap neuron, hitung perbaikan bobot dan bias dengan Persamaan 2.24 dan 2.25,

$$v_{ij}baru = v_{ij}lama + \Delta v_{ij}$$

$$v_{ij}baru = -0,30408 + 0$$

$$v_{ij}baru = -0,30408$$

$$b_{1j}baru = b_{1j}lama + \Delta b_{1j}$$

$$b_{1j}baru = 0,35601 + 0,010566$$

$$b_{1j}baru = 0,36658$$

Tabel 4.15 Hasil perubahan bobot untuk data latih-1

| LI -> LH | | | | | | | | | | | LH -> LO |
|----------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|-------------|
| w11 | w21 | w31 | w41 | w51 | w61 | w71 | w81 | w91 | w101 | b11 | v11 |
| -0,30408 | 0,17351 | 0,25946 | 0,24947 | 0,00640 | 0,28118 | 0,08987 | 0,47883 | 0,20878 | 0,48259 | 0,36658 | 0,525732278 |
| w12 | w22 | w32 | w42 | w52 | w62 | w72 | w82 | w92 | w102 | b12 | v21 |
| -0,07211 | 0,12292 | 0,09850 | 0,48145 | 0,10701 | 0,34265 | 0,20678 | 0,25208 | 0,18237 | 0,39343 | 0,46575 | 0,269659689 |
| | | | | | | | | | | | b21 |
| | | | | | | | | | | | 0,50022335 |

6. Hitung RMSE setiap *epoch* dengan Persamaan 2.28
7. Lakukan langkah diatas sebanyak data latih dan berhenti sesuai aturan yang ditetapkan seperti banyak *epoch* dan target error.

4.4.3.2 Testing Backpropagation

Tabel 4.16 Bobot akhir

| LI -> LH | | | | | | | | | | | LH -> LO |
|------------------|--------------|--------------|------------------|--------------|------------------|--------------|------------------|------------------|------------------|--------------|------------------|
| w11 | w21 | w31 | w41 | w51 | w61 | w71 | w81 | w91 | w101 | b11 | v11 |
| - 0,336 54 | 0,277 741 | 0,317 972 | - 0,217 96 | 0,002 628 | 0,454 661 | 0,170 138 | - 0,389 78 | 0,395 435 | 0,537 99 | 0,520 776 | 0,831 995 |
| w12 | w22 | w32 | w42 | w52 | w62 | w72 | w82 | w92 | w102 | b12 | v21 |
| - 0,072 92 | 0,121 376 | 0,097 852 | 0,478 336 | 0,106 708 | - 0,341 07 | 0,206 972 | - 0,254 07 | - 0,184 36 | - 0,394 05 | 0,462 941 | - 0,005 47 |
| | | | | | | | | | | | b21 |
| | | | | | | | | | | | 1,100 223 |

Langkah-langkah manualisasi proses *testing Backpropagation* adalah sebagai berikut.

1. Masukkan data PJK yang digunakan untuk proses *testing*. Data PJK dalam penelitian ini memiliki sepuluh fitur,

Tabel 4.17 Sample data uji

| Fitur Data | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---------------|---|----------|------|----------|---------|----------|----------|---|---|----|
| 1 | 1 | 0,086957 | 1 | 0,84375 | 0,34375 | 0,622768 | 1 | 1 | 1 | 0 |
| 2 | 1 | 0,478261 | 0,4 | 0,765625 | 0,4125 | 0,357143 | 0 | 1 | 1 | 0 |
| 3 | 1 | 0,391304 | 0,26 | 0,375 | 0,23125 | 0,116071 | 0,088095 | 0 | 1 | 0 |
| 4 | 0 | 0,347826 | 0,56 | 0 | 0,61875 | 0,149554 | 0,1 | 0 | 0 | 1 |

2. Untuk setiap neuron yang terdapat pada *layer* ke-*j*, hitung sinyal-sinyal berbobot dengan Persamaan 2.9, dan hitung sinyal *outputnya* dengan Persamaan 2.10, sinyal-sinyal tersebut selanjutnya diteruskan ke setiap neuron yang berada dalam *layer* diatasnya/selanjutnya,

$$z_{in_j} = b1_j + \sum_{i=1}^n x_i \times bobot_{ij}$$

$$z_{in_j} = 0,520776 + (1 \times -0,33654 + 0,086957 \times 0,277741 + \dots + 0 \times 0,53799)$$

$$z_{in_j} = 0,802306$$

$$z_j = f(z_{in_j})$$

$$z_j = \frac{1}{1 + e^{-z_{in_j}}}$$

$$z_j = \frac{1}{1 + e^{-0,802306}}$$

$$z_j = 0,690468$$

Layer output

$$y_{in_j} = b1_j + \sum_{i=1}^n x_i \times bobot_{ij}$$

$$y_{in_j} = 1,100223 + (0,690468 \times 0,831995 + \dots + 0 \times 0,53799)$$

$$y_{in_j} = 1,671288$$

$$y = f(y_{in_j})$$

$$y = bulatkan(1,671288)$$

$$y = 2$$

Tabel 4.18 Hasil uji

| Data | Hasil Prediksi | |
|------|----------------|---|
| 1 | 1,671288 | 2 |
| 2 | 1,604386 | 2 |
| 3 | 1,663326 | 2 |
| 4 | 1,767258 | 2 |

4.5 Perancangan User Interface

Dalam perancangan *user interface* memuat rancang tampilan dari beberapa *form* yang akan menjadi acuan dalam implementasi GUI program.

4.5.1 Form Depan

Rancangan *form* depan dapat dilihat pada Gambar 4.9. Dalam *form* depan hanya terdapat tiga tombol yang akan menampilkan jendela terkait. Penjelasan mengenai *form* ini adalah sebagai berikut.

1. Tombol *Scan Data*, tombol ini berfungsi untuk menampilkan jendela untuk memilih data yang akan dipindai oleh kedua metode untuk melakukan proses normalisasi.
2. Tombol *LVQ*, tombol untuk masuk ke dalam proses pelatihan metode LVQ.

Scan Data¹

LVQ²

Backpropagation³

Gambar 4.9 Form depan

3. Tombol *Backpropagation*, tombol untuk masuk ke dalam proses pelatihan metode *Backpropagation*.

4.5.2 Form Metode LVQ

Pilih Data Latih¹

1

Banyak Epoch 2

Learning rate 3

Pengurangan Learning rate 4

Minimum Learning rate 5

Latih⁶

Gambar 4.10 Form metode LVQ persiapan training

Rancangan *form* LVQ dapat dilihat pada Gambar 4.10 di atas. Form ini berfungsi memasukan parameter-parameter yang dibutuhkan dalam proses pelatihan LVQ. Berikut penjelasan form tersebut:

1. Tombol Pilih Data Latih untuk memilih data latih dan tabel untuk menampilkan data tersebut.
2. *Input number* untuk banyak *epoch*
3. *Input number* untuk *learning rate*.
4. *Input number* untuk pengurangan *learning rate*.
5. *Input number* untuk *learning rate* minimum.
6. Tombol Latih untuk memulai pelatihan.

4.5.3 Form Testing LVQ

Gambar 4.11 Form metode LVQ testing

Rancangan *form testing* LVQ dapat dilihat pada Gambar 4.11. Form ini berfungsi melakukan proses pengujian LVQ. Berikut penjelasan form tersebut:

1. Tombol untuk memilih data Uji dan tabel untuk menampilkan data terpilih.
2. Tombol uji untuk melakukan pengujian LVQ dan tabel yang akan menampilkan target aktual dan hasil prediksi.
3. *Numberfield* untuk menampilkan hasil eror.
4. *Numberfield* untuk menampilkan hasil akurasi.
5. Tombol untuk melihat grafik hasil pengujian.

4.5.4 Form Metode Backpropagation

Load Data Latih¹

1

Banyak neuron per hidden layer 2

Jenis fungsi aktivasi 3 ▼

Inisialisai jaringan⁷

Maksimum *Epoch* 4

Learning Rate 5

Target Error RMSE 6

Latih⁸

Gambar 4.12 Form metode Backpropagation

Rancangan *form training Backpropagation* dapat dilihat pada Gambar 4.12. Form ini berfungsi melakukan proses pelatihan *backpropagation* sesuai parameter yang dimasukkan. Berikut penjelasan form tersebut:

1. Tombol untuk memilih data latih dan tabel untuk menampilkan data terpilih.
2. *Input number* untuk banyak neuron per hidden layer. Contohnya: jika dimasukkan "4 5" akan dibuat dua hidden layer yang masing-masing memiliki neuron sebanyak empat dan lima.
3. *Dropdownmenu* untuk memilih jenis fungsi aktivasi.
4. *Input number* untuk *epoch* maksimum.
5. *Input number* untuk *learning rate*.
6. *Input number* untuk target eror RMSE.
7. Tombol untuk inisialisasi jaringan.
8. Tombol untuk memulai pelatihan.

4.5.5 Form Testing Backpropagation

The form is titled "Form Testing Backpropagation". It contains the following elements:

- A button labeled "Load Data Uji¹" at the top.
- A large rectangular area labeled "1" below the first button.
- A button labeled "Uji²" below area "1".
- Another large rectangular area labeled "2" below the "Uji²" button.
- Three output fields on the right side:
 - "Error" with a corresponding box labeled "3".
 - "Akurasi" with a corresponding box labeled "4".
 - "Lihat Grafik⁵" with a corresponding box.

Gambar 4.13 Form metode Backpropagation testing

Rancangan *form testing Backpropagation* dapat dilihat pada Gambar 4.13. Form ini berfungsi melakukan proses pengujian *backpropagation*. Berikut penjelasan form tersebut:

1. Tombol untuk memilih data Uji dan tabel untuk menampilkan data terpilih.
2. Tombol uji untuk melakukan pengujian dan tabel yang akan menampilkan target aktual dan hasil prediksi.
3. *Numberfield* untuk menampilkan hasil eror.
4. *Numberfield* untuk menampilkan hasil akurasi.
5. Tombol untuk melihat grafik hasil pengujian.

4.5.6 Form Hasil Pelatihan

Rancangan *form* yang digunakan untuk menampilkan hasil pelatihan untuk metode LVQ dan *backpropagation* dapat dilihat pada Gambar 4.14. Berikut penjelasan rancangan *form* tersebut:

1. *Numberfield* untuk menampilkan hasil eror/RMSE pelatihan dan tabel yang menampilkan hasil seluruh eror/RMSE per *epoch* yang terjadi saat pelatihan.

Gambar 4.14 *Form hasil pelatihan*

2. Tombol untuk melihat grafik hasil pengujian.
3. Tombol untuk masuk *form* pengujian.

4.6 Perancangan Uji Coba

Pengujian dan evaluasi pada penelitian ini digunakan untuk mengetahui kinerja program. Skenario pengujian yang dilakukan pada penelitian ini adalah:

1. Pengujian parameter *learning rate* untuk metode LVQ dan *Backpropagation*,
2. Pengujian parameter pengurangan *learning rate* untuk metode LVQ,
3. Pengujian pasangan data latih dan uji untuk metode LVQ dan *Backpropagation*,
4. Pengujian jumlah neuron untuk metode *Backpropagation*,
5. Pengujian jumlah *hidden layer* untuk metode *Backpropagation*,
6. Pengujian jenis fungsi aktivasi untuk metode *Backpropagation*,
7. *Cross validation* untuk menguji kinerja model LVQ dan *Backpropagation*.

4.6.1 Rancangan Pengujian Parameter *Learning Rate*

Rancangan pengujian ini bertujuan untuk mengetahui pengaruh peningkatan *learning rate* terhadap nilai akurasi dalam metode LVQ dan *Backpropagation*.

Tabel 4.19 Rancangan pengujian parameter *learning rate*

| Learning Rate | Nilai Akurasi Pelatihan dan Pengujian (%) Percobaan ke -i | | | | Rerata Akurasi | |
|---------------|--|-----------|-----------|-----------|----------------|-----------|
| | 1 | | n | | Pelatihan | Pengujian |
| | Pelatihan | Pengujian | Pelatihan | Pengujian | | |

Tabel 4.19 (lanjutan)

| | | | | | | |
|--|--|--|--|--|--|--|
| | | | | | | |
| | | | | | | |

4.6.2 Rancangan Pengujian Parameter Pengurangan *Learning Rate* Untuk LVQ

Rancangan pengujian ini bertujuan untuk mengetahui hubungan peningkatan pengurangan *learning rate* dalam metode LVQ terhadap nilai akurasi yang didapat.

Tabel 4.20 Rancangan pengujian parameter pengurangan *learning rate*

| Pengurangan Learning Rate | Nilai Akurasi Pelatihan dan Pengujian (%) Percobaan ke -i | | | | Rerata Akurasi | |
|---------------------------|---|-----------|-----------|-----------|----------------|-----------|
| | 1 | | n | | Pelatihan | Pengujian |
| | Pelatihan | Pengujian | Pelatihan | Pengujian | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |

4.6.3 Rancangan Pengujian Pasangan Data Latih dan Uji

Rancangan pengujian ini bertujuan untuk mengetahui pasangan data latih dan uji yang memiliki nilai akurasi yang terbaik baik saat proses pelatihan atau pengujian.

Tabel 4.21 Rancangan pengujian pasangan data latih dan uji

| Pasangan Data Latih dan Uji | Nilai Akurasi Pelatihan dan Pengujian (%) Percobaan ke -i | | | | Rerata Akurasi | |
|-----------------------------|---|-----------|-----------|-----------|----------------|-----------|
| | 1 | | n | | Pelatihan | Pengujian |
| | Pelatihan | Pengujian | Pelatihan | Pengujian | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |

4.6.4 Rancangan Pengujian Jumlah Neuron Untuk *Backpropagation*

Rancangan pengujian ini bertujuan untuk mengetahui hubungan antara peningkatan jumlah neuron terhadap nilai akurasi yang didapat.

Tabel 4.22 Rancangan pengujian jumlah neuron

| Jumlah Neuron | Nilai Akurasi Pelatihan dan Pengujian (%) Percobaan ke -i | | | | Rerata Akurasi | |
|---------------|---|-----------|-----------|-----------|----------------|-----------|
| | 1 | | n | | Pelatihan | Pengujian |
| | Pelatihan | Pengujian | Pelatihan | Pengujian | | |
| | | | | | | |

4.6.5 Rancangan Pengujian Jumlah *Hidden Layer* Untuk *Backpropagation*

Rancangan pengujian ini bertujuan untuk mengetahui hubungan antara peningkatan jumlah *hidden layer* terhadap nilai akurasi yang didapat.

Tabel 4.23 Rancangan pengujian jumlah *hidden layer*

| Jumlah Hidden Layer | Nilai Akurasi Pelatihan dan Pengujian (%) Percobaan ke -i | | | | Rerata Akurasi | |
|------------------------|--|-----------|-----------|-----------|----------------|-----------|
| | 1 | | n | | Pelatihan | Pengujian |
| | Pelatihan | Pengujian | Pelatihan | Pengujian | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |

4.6.6 Rancangan Pengujian Jenis Fungsi Aktivasi Untuk *Backpropagation*

Rancangan pengujian ini bertujuan untuk mencari tahu jenis fungsi aktivasi yang memiliki nilai akurasi terbaik.

Tabel 4.24 Rancangan pengujian jenis fungsi aktivasi

| Jenis Fungsi Aktivasi | Arsitektur <i>Hidden Layer</i> | Nilai Akurasi Pelatihan dan Pengujian (%) Percobaan ke -i | | | | Rerata Akurasi | |
|-----------------------------|-----------------------------------|--|-----------|-----------|-----------|----------------|-----------|
| | | 1 | | 10 | | Pelatihan | Pengujian |
| | | Pelatihan | Pengujian | Pelatihan | Pengujian | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |

4.6.7 *Cross Validation* (CV)

Pada Tabel 4.19 menggambarkan rancangan pengujian CV. Pengujian CV pada penelitian ini akan membagi 130 data yang dimiliki menjadi k buah *fold* data. *Fold* data tersebut digunakan untuk membandingkan akurasi dari metode LVQ dengan *Backpropagation* dengan parameter optimal dari hasil pengujian parameter kedua metode.

Tabel 4.25 Rancangan pengujian *cross validation*

| Fold-i \ Model-i | 1 | ... | N | Akurasi |
|------------------|---------------|---------------|---------------|---------|
| 1 | Data Testing | Data Training | Data Training | |
| ... | Data Training | Data Testing | Data Training | |
| 10 | Data Training | Data Training | Data Testing | |

BAB 5 IMPLEMENTASI

Bab Ini membahas tentang implementasi dari perancangan yang sudah dibuat sebelumnya yang mencakup arsitektur jaringan LVQ dan *Backpropagation*, algoritme pelatihan dan pengujian metode LVQ dan *Backpropagation*, dan antarmuka pengguna. Dalam proses implementasi ini digunakan bahasa pemrograman java dengan *tools* IDE Netbeans.

5.1 Implementasi Program

Implementasi program menjelaskan mengenai code yang telah dibuat berdasarkan perancangan dalam Bab sebelumnya. Berikut implementasi program yang dibuat.

5.1.1 Implementasi Normalisasi Data

Proses normalisasi data diawali dengan memindai nilai maksimum dan minimum pada sebuah data. Kode program proses memindai data dapat dilihat dalam Kode Program 5.1.

```

1  public void read() throws IOException {
2      File inputWorkbook = new File(inputFile);
3      Workbook w;
4
5      try {
6          w = Workbook.getWorkbook(inputWorkbook);
7          // Get the first sheet
8          Sheet sheet = w.getSheet(0);
9
10         namaAtribut = new String[sheet.getColumns()];
11         dataTertinggi = new double[sheet.getColumns()];
12         dataTerendah = new double[sheet.getColumns()];
13
14         //Catatan baris ke-0 dianggap tak digunakan
15         data = new
16             double[sheet.getRows()][sheet.getColumns()];
17         // Loop over first 10 column and lines
18         for (int j = 0; j < sheet.getColumns(); j++) {
19             dataTertinggi[j] = -99999.0;
20             dataTerendah[j] = 99999.0;
21             for (int i = 0; i < sheet.getRows(); i++) {
22                 Cell cell = sheet.getCell(j, i);
23                 CellType type = cell.getType();
24                 if (type == CellType.LABEL) {
25                     namaAtribut[j] = cell.getContents();
26                 }
27                 if (type == CellType.NUMBER) {

```

```

28         data[i][j] = Double.parseDouble(
29             cell.getContents());
30         if (dataTertinggi[j] < data[i][j]) {
31             dataTertinggi[j] = data[i][j];
32         }
33         if (dataTerendah[j] > data[i][j]) {
34             dataTerendah[j] = data[i][j];
35         }
36     }
37 }
38 }

```

Kode Program 5.1 Read data

Kode Program 5.1 adalah kode program *method read* yang berfungsi untuk mencari dan melakukan inisialisasi nilai maksimum dan nilai minimum dari suatu data. Berikut penjelasan dari Kode Program 5.1:

1. Baris kedua, Instansiasi variabel *inputWorkbook* yang bertipe File. Tipe file berasal dari *library* dalam jdk 1.7. Variabel ini akan menyimpan file excel yang ingin disimpan. Dalam instansiasi ini terdapat argumen *inputFile* bertipe String yang merepresentasikan direktori *file excel* yang disimpan dalam variabel ini.
2. Baris ketiga, deklarasi variabel *w* bertipe *workbook*. Tipe *workbook* berasal dari *library xlsx* (*library* untuk membaca *file excel*).
3. Baris keenam, insialisasi variabel *w* untuk menyimpan file dari variabel baris pertama.
4. Baris ke-10 sampai 12, instansiasi panjang variabel array *namaAtribut*, *dataTertinggi*, dan *dataTerendah* sebanyak kolom dalam file excel baris pertama.
5. Baris kelimabelas, instansiasi panjang variabel array *data* sebanyak jumlah baris dan kolom dalam file excel baris pertama.
6. Baris ke-18 sampai 37 memindai file baris kesatu untuk mendapat nilai *dataTertinggi* dan *dataTerendah* setiap kolomnya, serta mendapat data dari file baris pertama.

Setelah proses pemindaan selesai dilakukan, proses normalisasi dapat dilakukan. Kode Program proses normalisasi dapat dilihat dalam Kode Program 5.2.

```

1     public void normalisasi() {
2         for (int i = 0; i < data.length; i++) {
3             for (int j = 0; j < data[i].length - 1; j++) {
4                 //Normal Normalisasi (0,1)
5                 data[i][j] = (data[i][j] - dataTerendah[j])
6                     / (dataTertinggi[j] - dataTerendah[j]);
7             }
8         }
9     }

```

Kode Program 5.2 Normalisasi data

Kode Program 5.2 adalah *method* normalisasi yang digunakan untuk melakukan normalisasi suatu data. Penjelasan Kode Program 5.2 adalah sebagai berikut.

1. Baris ke-2 sampai 8 merupakan proses normalisasi data berdasarkan Persamaan 2.29.

5.1.2 Implementasi Membuat Jaringan LVQ

Proses pertama sebelum dapat melakukan pelatihan dan pengujian LVQ adalah membuat jaringannya. Kode program membuat jaringan LVQ dapat dilihat dalam Kode Program 5.3.

```

1  double[][] bobot;
2  public LVQ(String path) {
3      d = new DataManagement();
4      d.setInputFile(path);
5      try {
6          d.read();
7          int banyakFitur = d.namaAtribut.length - 1;
8          int banyakKlas = (int)
9              d.getDataTertinggi()[d.data[0].length - 1] + 1;
10         bobot = new double[banyakKlas][banyakFitur];
11     } catch (IOException ex) {
12         Logger.getLogger(LVQ.class.getName()).log(Level.SEVERE,
13             null, ex);
14     }
15 }
16
17 private void inisialisasiBobotAwal(DataManagement dataLatih) {
18     Random r = new Random();
19     int i = 0;
20     while (i <=
21         dataLatih.getDataTertinggi()[dataLatih.namaAtribut.length
22             - 1]) {
23         int random = r.nextInt(dataLatih.data.length - 1) + 1;
24
25         if (i ==
26             dataLatih.data[random][dataLatih.namaAtribut.length -
27                 1]) {
28             for (int j = 0; j < dataLatih.data[random].length
29                 - 1; j++) {
30                 bobot[i][j] = dataLatih.data[random][j];
31             }
32             i++;
33         }
34     }
35 }
36 }

```

Kode Program 5.3 Membuat jaringan LVQ

Kode Program 5.3 merupakan *sample* kode program kelas LVQ untuk membuat jaringan LVQ. Dalam kode program tersebut terdapat satu variabel global, sebuah konstruktor, dan sebuah *method*. Berikut penjelasan Kode Program 5.3:

1. Baris pertama merupakan deklarasi variabel array2D bobot bertipe double. Variabel ini adalah representasi untuk banyak bobot dalam suatu *node* atau neuron.
2. Baris ke-2 sampai 15 adalah konstruktor LVQ yang digunakan untuk membuat jaringan LVQ sesuai dengan data yang ingin dilatih dan diuji.
3. Baris ke-17 sampai 36 adalah *method* inisialisasiBobotAwal. *Method* ini digunakan untuk memberikan nilai awal bobot-bobot setiap neuron. *Method* ini akan dipanggil dalam *method* pelatihan LVQ.

5.1.3 Implementasi Pelatihan LVQ

Setelah Jaringan LVQ sudah dibuat barulah jaringan LVQ dilatih. Kode program proses pelatihan LVQ dapat dilihat dalam Kode Program 5.4. Kode Program 5.4 merupakan *method* *train* yang memiliki lima parameter, parameter *d* untuk data latih, parameter *maxEpoch* untuk maksimum *epoch*, parameter *learningRate* untuk nilai rasio pembelajaran(alpha), *decLearningRate* untuk pengecilan nilai alpha, dan *minLearningRate* untuk minimum alpha yang diperbolehkan. Berikut penjelasan dari Kode Program 5.4:

1. Baris ke-4 memanggil *method* inisialisasiBobotAwal.
2. Baris ke-6 sampai 13, inisialisasi dataLatih dari parameter *d*.
3. Baris ke-15 sampai 19, inisialisasi dataTarget dari parameter *d*.

```

1 public void train(DataManagement d, int maxEpoch, double
2     learningRate, double decLearningRate, double
3     minLearningRate) {
4     inisialisasiBobotAwal(d);
5
6     double[][] dataLatih = new double[d.data.length -
7         1][d.namaAtribut.length - 1];
8
9     for (int i = 0; i < dataLatih.length; i++) {
10         for (int j = 0; j < dataLatih[i].length; j++) {
11             dataLatih[i][j] = d.data[i + 1][j];
12         }
13     }
14     //inisialisasi target
15     double[] dataTarget = new double[d.data.length - 1];
16     for (int i = 0; i < dataTarget.length; i++) {
17         dataTarget[i] = d.data[i +
18             1][d.namaAtribut.length - 1];
19     }
20
21     logErrorPelatihan = new ArrayList<>();
22

```

```

23      /*Proses Pelatihan*/
24      int error = 1;
25      double[] jarakThdKlas = new double[(int)
26      d.getDataTertinggi()[d.namaAtribut.length - 1] + 1];
27      int epoch = 0;
28      while (epoch < maxEpoch && learningRate >
29      minLearningRate && error != 0) {
30          error = 0;
31          for (int i = 0; i < dataLatih.length; i++) {
32              for (int j = 0; j < jarakThdKlas.length;
33              j++) {
34                  jarakThdKlas[j] =
35                  hitungJarakEuclidean(dataLatih[i],
36                  bobot[j]);
37
38              }
39              int klasTerdekat =
40              cariKlasTerdekat(jarakThdKlas);
41
42              /*Update Bobot*/
43              if (klasTerdekat == dataTarget[i]) {
44                  for (int j = 0; j <
45                  bobot[klasTerdekat].length; j++) {
46                      bobot[klasTerdekat][j] =
47                      bobot[klasTerdekat][j] +
48                      learningRate * (dataLatih[i][j] -
49                      bobot[klasTerdekat][j]);
50                  }
51              } else {
52                  error++;
53                  for (int j = 0; j <
54                  bobot[klasTerdekat].length; j++) {
55                      bobot[klasTerdekat][j] =
56                      bobot[klasTerdekat][j] -
57                      learningRate * (dataLatih[i][j] -
58                      bobot[klasTerdekat][j]);
59                  }
60              }
61          }
62
63      }
64
65      logErrorPelatihan.add(error);
66
67      learningRate = learningRate - (learningRate *
68      decLearningRate);
69
70      epoch++;
71  }
72
73  }

```

Kode Program 5.4 Proses pelatihan LVQ

4. Baris ke-21 adalah instansiasi variabel `logErrorPelatihan` yang digunakan untuk menyimpan error setiap *epoch*(perulangan) pelatihan.
5. Baris ke 24 sampai 71 adalah representasi dari proses pelatihan LVQ berdasarkan digram alir dalam Gambar 4.5.

5.1.4 Implementasi Pengujian LVQ

Setelah proses pelatihan LVQ selesai dilakukan, proses pengujian LVQ dilakukan. Kode program proses pengujian LVQ dapat dilihat dalam Kode Program 5.5. Kode Program 5.5 adalah *method test* yang memiliki satu parameter `d` yang merepresentasikan data uji. Berikut penjelasan dari Kode Program 5.5:

1. Baris ke-3 sampai 9 adalah inisialisasi `dataUji` berdasarkan parameter `d`.
2. Baris ke-10, instansiasi variabel `array2d` hasil sebanyak `dataUji`. Variabel ini digunakan untuk menyimpan hasil pengujian.

```

1 public void test(DataManagement d) {
2     //inisialisasi data uji
3     double[][] dataUji = new double[d.data.length -
4     1][d.namaAtribut.length - 1];
5     for (int i = 0; i < dataUji.length; i++) {
6         for (int j = 0; j < dataUji[i].length; j++) {
7             dataUji[i][j] = d.data[i + 1][j];
8         }
9     }
10    this.hasil = new double[dataUji.length][2];
11    //inisialisasi target
12    double[] dataTarget = new double[d.data.length - 1];
13    for (int i = 0; i < dataTarget.length; i++) {
14        dataTarget[i] = d.data[i +
15        1][d.namaAtribut.length - 1];
16    }
17    int error = 0;
18    double[] jarakThdKlas = new double[(int)
19    d.getDataTertinggi() [d.namaAtribut.length - 1] + 1];
20    for (int i = 0; i < dataUji.length; i++) {
21        for (int j = 0; j < jarakThdKlas.length; j++) {
22            jarakThdKlas[j] =
23            hitungJarakEuclidean(dataUji[i], bobot[j]);
24
25        }
26
27        int klasTerdekat =
28        cariKlasTerdekat(jarakThdKlas);
29        if (klasTerdekat != dataTarget[i]) {
30            error++;
31        }
32
33        this.hasil[i][0] = dataTarget[i];
34        this.hasil[i][1] = klasTerdekat;

```

```

35         }
36
37         this.error = error;
38         this.akurasi = (dataUji.length - this.error) /
39             dataUji.length * 100;
40     }

```

Kode Program 5.5 Proses pengujian LVQ

3. Baris ke-12 sampai 16 adalah inialisasi target berdasarkan parameter d.
4. Baris ke-17 sampai 39 merepresentasikan proses pengujian LVQ berdasarkan diagram alir Gambar 4.6.

5.1.5 Implementasi Membuat Jaringan *Backpropagation*

Proses pertama sebelum dapat melakukan pelatihan dan pengujian *backpropagation* adalah membuat jaringannya. Kode program membuat jaringan *backpropagation* dapat dilihat dalam Kode Program 5.6.

```

1  ArrayList<double[][]> bobot;
2  int[] banyakLDanNeuron;
3
4  public Backpropagation(int[] nLnN, String path) {
5      d = new DataManagement();
6      bobot = new ArrayList<>();
7      banyakLDanNeuron = nLnN;
8      d.setInputFile(path);
9      try {
10         d.read();
11         d.normalisasi();
12     } catch (IOException ex) {
13         Logger.getLogger(
14             Backpropagation.class.getName()).log(Level.SEVERE,
15             null, ex);
16     }
17
18     /*Pembentukan Layer dan Neuronnya*/
19     for (int i = 0; i < nLnN.length; i++) {
20         if (i == 0) {
21             bobot.add(new
22                 double[nLnN[0]][d.namaAtribut.length]);
23         } else {
24             bobot.add(new double[nLnN[i]][nLnN[i - 1] +
25                 1]);
26         }
27     }
28
29     /*Inisialisasi bobot secara random*/
30     Random r = new Random();
31     for (int i = 0; i < bobot.size(); i++) {
32         for (int j = 0; j < bobot.get(i).length; j++) {
33             for (int k = 0; k < bobot.get(i)[j].length;
34                 k++) {
35                 bobot.get(i)[j][k] = 0 + (0 + 0.5) *
36                     r.nextDouble();
37             }
38         }
39     }

```

```
39     }
40 }
```

Kode Program 5.6 Membuat jaringan *Backpropagation*

Kode Program 5.6 adalah *sample* kode program kelas *Backpropagation* untuk membuat jaringannya, Dalam kode program tersebut terdapat dua variabel global dan sebuah konstruktor. Berikut penjelasan Kode Program 5.6:

1. Baris pertama dan kedua adalah deklarasi variabel global bobot dan banyakLDanNeuron yang merepresentasikan banyaknya bobot per nueron dalam setiap layer.
2. Baris ke-4 sampai 40 adalah konstruktor *Backpropagation* yang digunakan untuk membuat jaringan *Backpropagation* sesuai dengan data yang ingin dilatih dan diuji serta melakukan inisialisasi bobot secara random dalam range(0, 0.5).

5.1.6 Implementasi Pelatihan *Backpropagation*

Setelah Jaringan *Backpropagation* sudah dibuat barulah jaringan *Backpropagation* dilatih. Kode program proses pelatihan *Backpropagation* dapat dilihat dalam Kode Program 5.7. Kode Program 5.7 merupakan *method train* yang memiliki empat parameter, parameter *d* untuk data latih, parameter *maxEpoch* untuk maksimum *epoch*, parameter *learningRate* untuk nilai rasio pembelajaran(alpha), *targetError* untuk nilai eror minimum yang ingin dicapai.

```
1 public void train(DataManagement d, int maxEpoch, double
2 learningRate, double targetError) {
3     //inisialisasi data latih
4     double[][] dataLatih = new double[d.data.length -
5 1][d.namaAtribut.length];
6     for (int i = 0; i < dataLatih.length; i++) {
7         for (int j = 0; j < dataLatih[i].length; j++) {
8             dataLatih[i][j] = d.data[i + 1][j];
9             if (j == 10) {
10                 dataLatih[i][j] = 1;
11             }
12         }
13     }
14
15
16
17     //inisialisasi target
18     double[] dataTarget = new double[d.data.length - 1];
19     for (int i = 0; i < dataTarget.length; i++) {
20         dataTarget[i] = d.data[i + 1][d.namaAtribut.length -
21 1];
22     }
23
24
25     logError = new ArrayList<>();
26     logRMSE = new ArrayList<>();
27
28     /*Proses Pelatihan*/
```

```

29 double rmse = 1;
30 int epoch = 0;
31 int erorPerEpoch;
32 while (epoch < maxEpoch && rmse > targetError) {
33     double totalError = 0;
34     erorPerEpoch = 0;
35     for (int i = 0; i < dataLatih.length; i++) {
36         ArrayList<double[]> signalInPerNeuron = new
37             ArrayList<>();
38         ArrayList<double[]> signalOutPerNeuron = new
39             ArrayList<>();
40         double error = 0;
41         /*Langkah Maju*/
42         for (int j = 0; j < bobot.size(); j++) {
43             double[] signalIn;
44             double[] signalOut;
45             if (j == bobot.size() - 1) {
46                 signalIn = new
47                     double[bobot.get(j).length];
48                 signalOut = new
49                     double[bobot.get(j).length];
50             } else {
51                 signalIn = new double[bobot.get(j).length
52                     + 1];
53                 signalOut = new
54                     double[bobot.get(j).length + 1];
55             }
56             for (int k = 0; k < bobot.get(j).length; k++) {
57                 for (int l = 0; l <
58                     bobot.get(j)[k].length; l++) {
59                     if (j == 0) {
60                         signalIn[k] =
61                             hitungSignal(dataLatih[i],
62                                 bobot.get(j)[k]);
63                         signalOut[k] =
64                             fungsiAktivasi(signalIn[k],
65                                 fungsiAktivasi);
66                     } else if (j == bobot.size() - 1) {
67                         //untuk output layer
68                         signalIn[k] =
69                             hitungSignal(signalOutPerNeuron.
70                                 get(j - 1), bobot.get(j)[k]);
71                         signalOut[k] =
72                             fungsiAktivasi(signalIn[k], 2);
73                     } else {
74                         signalIn[k] =
75                             hitungSignal(signalOutPerNeuron.
76                                 get(j - 1), bobot.get(j)[k]);
77                         signalOut[k] =
78                             fungsiAktivasi(signalIn[k],
79                                 fungsiAktivasi);
80                     }
81                 }
82             }
83             if (j != bobot.size() - 1) {
84                 signalIn[signalIn.length - 1] = 1;
85                 signalOut[signalOut.length - 1] = 1;
86             }
87         }

```

```

88         if (j != bobot.size() - 1) {
89
90         }
91         signalInPerNeuron.add(signalIn);
92         signalOutPerNeuron.add(signalOut);
93
94     }
95
96     /*Langkah Mundur*/
97     ArrayList<double[][]> bobotMundur = new
98     ArrayList<>();
99     for (int j = bobot.size() - 1; j > -1; j--) {
100         if (j != 0) {
101
102             bobotMundur.add(new double[bobot.get(j) -
103             1].length + 1][bobot.get(j).length]);
104         } else {
105
106             bobotMundur.add(new
107             double[d.namaAtribut.length][
108             bobot.get(j).length]);
109         }
110     }
111
112     int a = bobot.size() - 1;
113     for (int j = 0; j < bobotMundur.size(); j++) {
114         for (int k = 0; k <
115         bobotMundur.get(j).length; k++) {
116
117             for (int l = 0; l <
118             bobotMundur.get(j)[k].length; l++) {
119                 bobotMundur.get(j)[k][l] =
120                 bobot.get(a)[l][k];
121             }
122         }
123     }
124     a--;
125
126 }
127
128 ArrayList<double[]> koreksiBobotperNeuron = new
129 ArrayList<>();
130 ArrayList<double[]> deltaIn = new ArrayList<>();
131 double e = 0;
132 int iii = 0;
133
134 for (int j = bobot.size() - 1; j > -1; j--) {
135
136     double[] infError = new
137     double[bobot.get(j).length];
138
139     for (int k = 0; k < bobot.get(j).length; k++)
140     {
141         double[] korBot = new
142         double[bobot.get(j)[k].length];
143
144         for (int l = 0; l <
145         bobot.get(j)[k].length; l++) {
146             if (j == bobot.size() - 1) {

```

```

147         e = dataTarget[i] -
148             signalOutPerNeuron.get(j)[k];
149
150         //Untuk Menghitung RMSE
151         eror = e * e;
152
153         infError[k] = e *
154             fungsiTurunan(signalInPerNeuron.
155                 get(j)[k], 10);
156         if (banyakLDanNeuron.length == 1)
157         {
158             korBot[l] = learningRate *
159                 infError[k] *
160                 dataLatih[i][l];
161         } else {
162             korBot[l] = learningRate *
163                 infError[k] *
164                 signalOutPerNeuron.get(j -
165                     1)[l];
166         }
167     } else if (j == 0) {
168         e = hitungSignal(deltaIn.get(
169             signalOutPerNeuron.size() - 2 -
170             j), bobotMundur.
171             get(signalInPerNeuron.
172                 size() - 2 - j)[k]);
173         infError[k] = e * fungsiTurunan(
174             signalInPerNeuron.get(j)[k],
175             fungsiAktivasi);
176         korBot[l] = learningRate *
177             infError[k] * dataLatih[i][l];
178     } else {
179         e = hitungSignal(
180             deltaIn.get(
181                 signalOutPerNeuron.size() - 2 -
182                 j), bobotMundur.get(
183                 signalInPerNeuron.size() - 2 -
184                 j)[k]);
185         infError[k] = e * fungsiTurunan(
186             signalInPerNeuron.get(j)[k],
187             fungsiAktivasi);
188
189         korBot[l] = learningRate *
190             infError[k] *
191             signalOutPerNeuron.get(j -
192                 1)[l];
193     }
194     koreksiBobotperNeuron.add(korBot);
195 }
196 deltaIn.add(infError);
197 }
198 if (e != 0) {
199     erorPerEpoch++;
200 }

```



```

205         }
206
207         /*Perbarui Bobot*/
208         int count = koreksiBobotperNeuron.size() - 1;
209         for (int j = 0; j < bobot.size(); j++) {
210             for (int k = bobot.get(j).length - 1; k > -1;
211                 k--) {
212                 for (int l = 0; l <
213                     bobot.get(j)[k].length; l++) {
214                     bobot.get(j)[k][l] =
215                         bobot.get(j)[k][l] +
216                         koreksiBobotperNeuron.get(count)[l];
217                 }
218                 count--;
219             }
220         }
221         totalError += eror;
222     }
223     totalError = totalError / dataLatih.length;
224     rmse = Math.sqrt(totalError);
225     logRMSE.add(rmse);
226     logError.add(erorPerEpoch);
227     epoch++;
228 }
229 }
230 }
231 }

```

Kode Program 5.7 Proses pelatihan *Backpropagation*

Penjelasan mengenai Kode Program 5.7 adalah sebagai berikut.

1. Baris ke-4 sampai 15, inisialisasi *dataLatih* dari parameter *d*.
2. Baris ke-18 sampai 23, insialisasi *dataTarget* dari parameter *d*.
3. Baris ke-25 dan 26 adalah instansiasi variabel *logError* dan *logRMSE* yang digunakan untuk menyimpan eror dan RMSE setiap *epoch*(perulangan) pelatihan.
4. Baris ke 29 sampai 228 adalah representasi dari proses pelatihan *Backpropagation* berdasarkan digram alir dalam Gambar 4.7 mulai dari pelatihan maju sampai pelatihan mundur.

5.1.7 Implementasi Pengujian *Backpropagation*

Setelah proses pelatihan *Backpropagation* selesai dilakukan, proses pengujian *Backpropagation* dilakukan. Kode program proses pengujian *Backpropagation* dapat dilihat dalam Kode Program 5.8.

```

1 public void test(DataManagement d) {
2     //inisialisasi data uji
3     double[][] dataUji = new double[d.data.length -
4         1][d.namaAtribut.length];
5     for (int i = 0; i < dataUji.length; i++) {
6         for (int j = 0; j < dataUji[i].length; j++) {
7             dataUji[i][j] = d.data[i + 1][j];
8             if (j == 10) {
9                 dataUji[i][j] = 1;

```

```

10         }
11     }
12 }
13
14 //inisialisasi target
15 double[] dataTarget = new double[d.data.length - 1];
16 for (int i = 0; i < dataTarget.length; i++) {
17     dataTarget[i] = d.data[i +
18         1][d.namaAtribut.length - 1];
19 }
20
21 this.hasil = new double[dataUji.length][2];
22 this.hasil[0] = dataTarget;
23 /*Proses Pengujian*/
24 double rmse = 1;
25 int epoch = 0;
26 double[] nilai = new double[dataUji.length];
27 double[] prediksi = new double[dataUji.length];
28 double totalError = 0;
29 int error = 0;
30 for (int i = 0; i < dataUji.length; i++) {
31     ArrayList<double[]> signalInPerNeuron = new
32         ArrayList<>();
33     ArrayList<double[]> signalOutPerNeuron = new
34         ArrayList<>();
35     /*Langkah Maju*/
36     for (int j = 0; j < bobot.size(); j++) {
37         double[] signalIn;
38         double[] signalOut;
39         if (j == bobot.size() - 1) {
40             signalIn = new
41                 double[bobot.get(j).length];
42             signalOut = new
43                 double[bobot.get(j).length];
44         } else {
45             signalIn = new
46                 double[bobot.get(j).length + 1];
47             signalOut = new
48                 double[bobot.get(j).length + 1];
49         }
50         for (int k = 0; k < bobot.get(j).length;
51             k++) {
52             for (int l = 0; l <
53                 bobot.get(j)[k].length; l++) {
54                 if (j == 0) {
55                     signalIn[k] =
56                         hitungSignal(dataUji[i],
57                             bobot.get(j)[k]);
58                     signalOut[k] =
59                         fungsiAktivasi(signalIn[k],
60                             fungsiAktivasi);
61                 } else if (j == bobot.size() - 1) {
62                     //untuk output layer
63                     signalIn[k] = hitungSignal(
64                         signalOutPerNeuron.get(j - 1),
65                         bobot.get(j)[k]);
66                     signalOut[k] =
67                         fungsiAktivasi(signalIn[k], 2);
68                     nilai[i] = signalIn[k];

```

```

69         prediksi[i] = signalOut[k];
70     } else {
71         signalIn[k] = hitungSignal(
72             signalOutPerNeuron.get(j - 1),
73             bobot.get(j)[k]);
74         signalOut[k] =
75             fungsiAktivasi(signalIn[k],
76                             fungsiAktivasi);
77     }
78 }
79 if (j != bobot.size() - 1) {
80     signalIn[signalIn.length - 1] = 1;
81     signalOut[signalOut.length - 1] = 1;
82 }
83 }
84 signalInPerNeuron.add(signalIn);
85 signalOutPerNeuron.add(signalOut);
86 }
87 if (prediksi[i] != dataTarget[i]) {
88     eror++;
89 }
90 }
91 this.error = eror;
92 this.akurasi = (dataTarget.length - this.error) /
93     dataTarget.length * 100;
94 this.hasil[1] = prediksi;
95 }

```

Kode Program 5.8 Proses pengujian *Backpropagation*

Kode Program 5.8 adalah *method test* yang memiliki satu parameter *d* yang merepresentasikan data uji. Berikut penjelasan dari Kode Program 5.8:

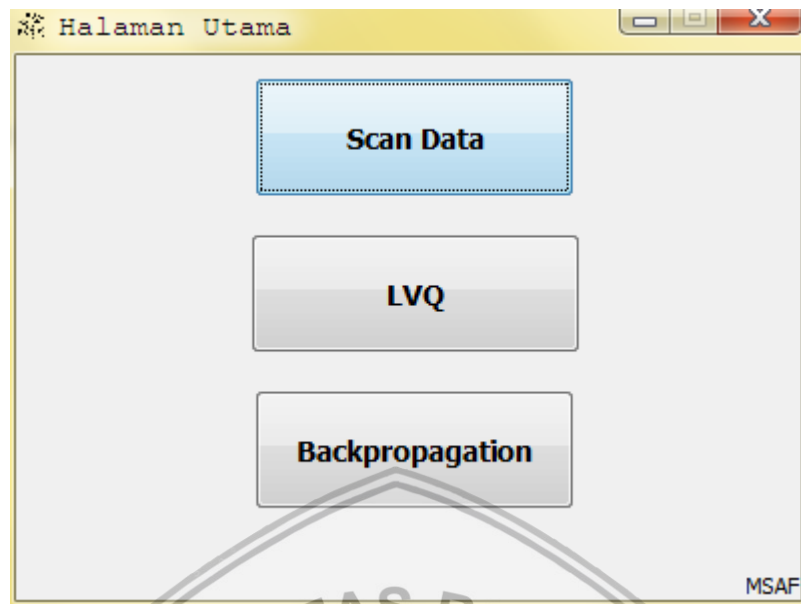
1. Baris ke-3 sampai 12 adalah inialisasi dataUji berdasarkan parameter *d*.
2. Baris ke-15 sampai 19 adalah inialisasi target berdasarkan parameter *d*.
3. Baris ke-21, instansiasi variabel array2d hasil sebanyak dataUji. Variabel ini digunakan untuk menyimpan hasil pengujian.
4. Baris ke-24 sampai 94 merepresentasikan proses pengujian *Backpropagation* berdasarkan diagram alir Gambar 4.8.

5.2 Hasil Implementasi Antarmuka Pengguna

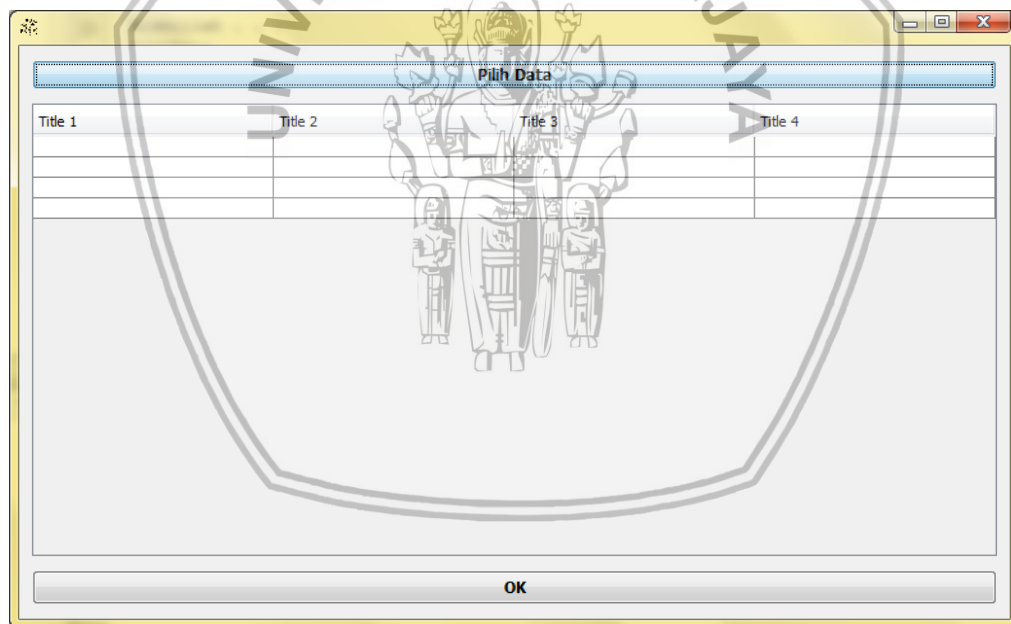
Hasil dari implementasi perancangan antarmuka pengguna adalah sebagai berikut.

5.2.1 Form Depan

Form depan adalah form yang memiliki tiga menu. Menu yang pertama adalah *Scan Data*, menu ini akan menampilkan *Form* seperti Gambar 5.2 yang digunakan untuk memilih data yang akan dijadikan referensi proses normalisasi data, Menu yang kedua adalah *LVQ*, menu ini akan menampilkan *Form* Gambar 5.3 yakni *Form* metode *LVQ*. Menu yang ketiga adalah *Backpropagation*, menu ini akan menampilkan *Form* Gambar 5.5 yakni *Form* metode *backpropagation*.



Gambar 5.1 Form depan



Gambar 5.2 Form scan data

Form scan data memiliki tiga komponen utama yaitu tombol pilih data, sebuah tabel, dan tombol ok. Tombol pilih data akan menampilkan *jfilechooser* yang digunakan untuk memilih file berekstensi .xls. Data dalam file tersebut selanjutnya akan ditampilkan pada tabel. Tombol ok akan menyimpan file tersebut. File tersebut akan digunakan sebagai referensi proses normalisasi data saat proses pelatihan dan pengujian kedua metode.

5.2.2 Form Metode LVQ

Form metode LVQ adalah form pelatihan LVQ. Form ini berfungsi untuk memasukkan data latih, dan parameter-parameter yang diperlukan untuk melakukan pelatihan LVQ. Jika data latih dan parameter-parameter sudah dimasukkan proses pelatihan dapat dimulai dengan menekan tombol Latih. Setelah proses pelatihan selesai akan tampil Form Gambar 5.7. Form metode LVQ dapat dilihat pada Gambar 5.3.

Gambar 5.3 Form metode LVQ

5.2.3 Form Testing LVQ

Form testing LVQ Gambar 5.4 adalah form yang berfungsi untuk melakukan proses pengujian LVQ. Form ini akan menampilkan data uji yang dipilih dan hasil dari pengujian LVQ yang dilakukan yaitu hasil prediksi, tingkat eror, dan akurasi. Tombol lihat grafik akan menampilkan grafik antara hasil prediksi dengan data aktual.

Gambar 5.4 Form testing LVQ

5.2.4 Form Metode Backpropagation

Form metode *backpropagation* adalah form pelatihan *Backpropagation*. Form ini berfungsi untuk memasukkan data latih, dan parameter-parameter yang diperlukan untuk melakukan pelatihan *backpropagation*. Jika data latih dan parameter-parameter sudah dimasukkan maka proses pelatihan dapat dimulai dengan menekan tombol Latih. Setelah proses pelatihan selesai akan tampil Form Gambar 5.8. Form metode *Backpropagation* dapat dilihat pada Gambar 5.5.

Gambar 5.5 Form metode Backpropagation

5.2.5 Form Testing Bacpropagation

Penguujian Backpropagation

Data Uji Pilih Data Uji

| Title 1 | Title 2 | Title 3 | Title 4 |
|---------|---------|---------|---------|
| | | | |
| | | | |
| | | | |

Uji

| Title 1 | Title 2 | Title 3 | Title 4 | Error |
|---------|---------|---------|---------|-------|
| | | | | |
| | | | | |
| | | | | |

Akurasi

Lihat Grafik

Gambar 5.6 Form testing Backpropagation

Form testing Backpropagation Gambar 5.6 adalah form yang berfungsi untuk melakukan proses pengujian *Backpropagation*. Form ini akan menampilkan data uji yang dipilih dan hasil dari pengujian *Backpropagation* yang dilakukan yaitu hasil prediksi, tingkat eror, dan akurasi. Tombol lihat grafik akan menampilkan grafik antara hasil prediksi dengan data aktual.

5.2.6 Form Hasil Pelatihan

Hasil Pelatihan

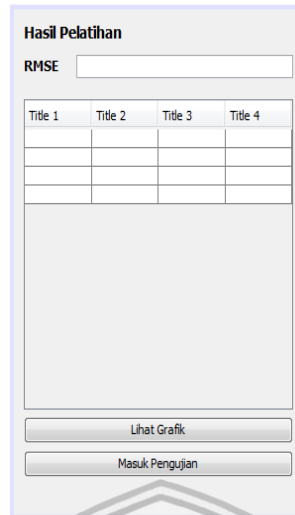
Error

| Title 1 | Title 2 | Title 3 | Title 4 |
|---------|---------|---------|---------|
| | | | |
| | | | |
| | | | |

Lihat Grafik

Masuk Pengujian

Gambar 5.7 Form hasil pelatihan LVQ



Hasil Pelatihan

RMSE

| Title 1 | Title 2 | Title 3 | Title 4 |
|---------|---------|---------|---------|
| | | | |
| | | | |
| | | | |
| | | | |

Gambar 5.8 Form hasil pelatihan *Backpropagation*

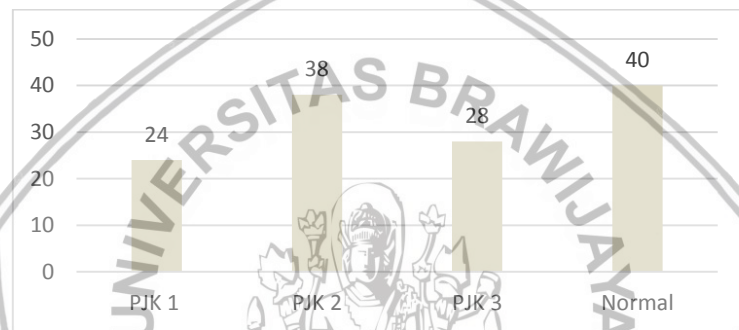
Form hasil pelatihan Gambar 5.7 dan 5.8 adalah form yang tampil setelah proses pelatihan selesai dilakukan. *form* ini menampilkan eror atau RMSE yang dihasilkan oleh proses pelatihan. Tabel dalam *form* ini berfungsi untuk menampilkan eror atau RMSE setiap *epoch*nya. Tombol Lihat grafik akan menampilkan grafik eror atau RMSE setiap *epoch*nya. Tombol masuk pengujian akan menampilkan *form* Gambar 5.4 untuk LVQ dan Gambar 5.6 untuk *Backpropagation*.

BAB 6 HASIL DAN PEMBAHASAN

Bab ini menjelaskan mengenai hasil analisis data, pengujian dan pembahasan dari uji coba yang telah dilakukan. Hal ini bertujuan untuk memperoleh perbandingan akurasi kinerja antara metode LVQ dengan *Backpropagation* dalam prediksi dini penyakit jantung koroner.

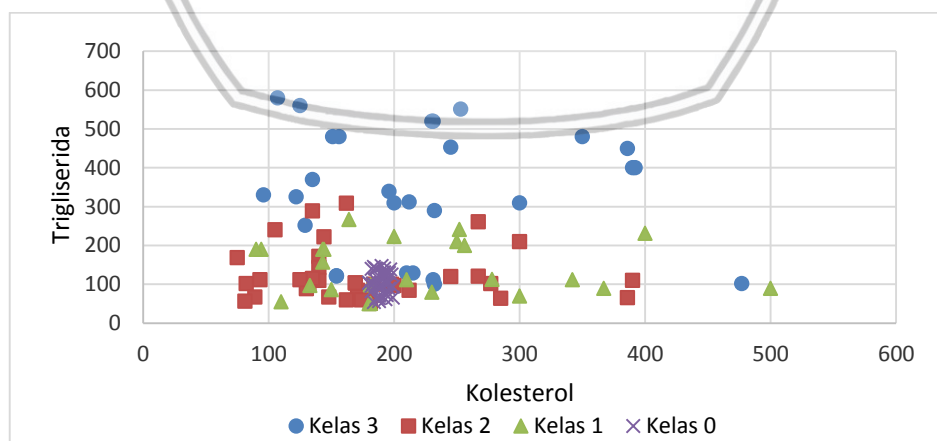
6.1 Analisis Data Deskriptif

Dalam penelitian ini digunakan data penyakit jantung koroner yang berjumlah 130 data dengan sepuluh parameter. Data tersebut terbagi menjadi empat kelas, yaitu PJK Tipe 3 (Kelas 3), PJK Tipe 2 (Kelas 2), PJK Tipe 1 (Kelas 1), Normal (Kelas 0). Gambar 6.1 menunjukkan diagram jumlah data setiap kelasnya.



Gambar 6.1 Diagram jumlah data setiap kelas

Untuk mendapatkan gambaran umum sebaran dari data tersebut maka dibuatlah *scatter plot*. Salah satu *scatter plot* dengan parameter kolesterol dan trigliserida dapat dilihat pada Gambar 6.2. (*scatter plot* untuk parameter lainnya dapat dilihat di lampiran A).



Gambar 6.2 *Scatter plot* kolesterol dengan trigliserida

Dari Gambar 6.2 menunjukkan bahwa pola penyebaran kelas sangat acak dan tidak beraturan serta jarak setiap kelasnya sangat dekat sehingga data PJK dalam

penelitian ini tidak dapat diklasifikasikan secara linear. Pola tersebut biasa disebut dengan pola *nonlinear* dalam data klasifikasi.

6.2 Hasil dan Pembahasan Pengujian Parameter *Learning Rate* Untuk Metode LVQ

Pengujian ini bertujuan untuk mengetahui hubungan antara tingkat *learning rate* terhadap tingkat akurasi yang didapat, Tabel 6.1 dan 6.2 merupakan hasil percobaan tingkat *learning rate* terhadap akurasi (hasil lengkap dapat dilihat dalam lampiran C.1).

Tabel 6.1 Hasil pengujian parameter *learning rate* dalam metode LVQ dengan peningkatan 0,01

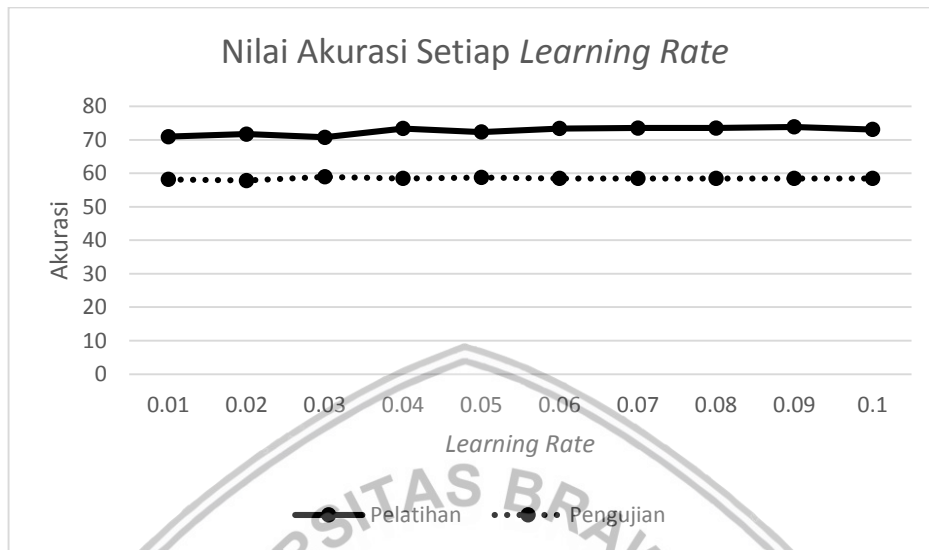
| <i>Learning rate</i> | Nilai Akurasi Pelatihan dan Pengujian (%) Percobaan ke -i | | | | Rerata Akurasi | |
|----------------------|--|-----------|-----------|-----------|----------------|-----------|
| | 1 | | 10 | | Pelatihan | Pengujian |
| | Pelatihan | Pengujian | Pelatihan | Pengujian | | |
| 0,01 | 72,308 | 58,462 | 72,308 | 58,462 | 70,923 | 58,154 |
| 0,02 | 70,769 | 58,462 | 73,846 | 58,462 | 71,692 | 57,846 |
| 0,03 | 66,154 | 56,923 | 67,692 | 60,000 | 70,769 | 58,923 |
| 0,04 | 73,846 | 58,462 | 73,846 | 58,462 | 73,385 | 58,462 |
| 0,05 | 73,846 | 58,462 | 72,308 | 58,462 | 72,308 | 58,769 |
| 0,06 | 73,846 | 58,462 | 73,846 | 58,462 | 73,385 | 58,462 |
| 0,07 | 73,846 | 58,462 | 73,846 | 58,462 | 73,538 | 58,462 |
| 0,08 | 72,308 | 58,462 | 73,846 | 58,462 | 73,538 | 58,462 |
| 0,09 | 73,846 | 58,462 | 73,846 | 58,462 | 73,846 | 58,462 |
| 0,1 | 73,846 | 58,462 | 72,308 | 58,462 | 73,077 | 58,462 |

Tabel 6.2 Hasil pengujian parameter *learning rate* dalam metode LVQ dengan peningkatan 0,1

| <i>Learning rate</i> | Nilai Akurasi Pelatihan dan Pengujian (%) Percobaan ke -i | | | | Rerata Akurasi | |
|----------------------|--|-----------|-----------|-----------|----------------|---------------|
| | 1 | | 10 | | Pelatihan | Pengujian |
| | Pelatihan | Pengujian | Pelatihan | Pengujian | | |
| 0,1 | 73,846 | 58,462 | 73,846 | 58,462 | 73,231 | 58,462 |
| 0,2 | 55,385 | 50,769 | 55,385 | 50,769 | 54,154 | 48,154 |
| 0,3 | 56,923 | 53,846 | 30,769 | 30,769 | 52,462 | 46,154 |
| 0,4 | 30,769 | 30,769 | 21,538 | 21,538 | 28,615 | 28,615 |
| 0,5 | 21,538 | 21,538 | 29,231 | 29,231 | 24,000 | 24,000 |

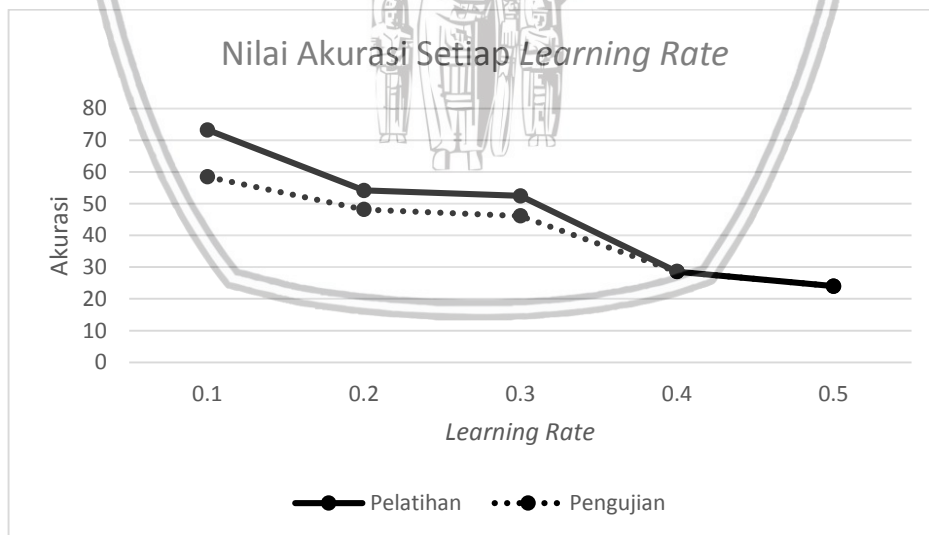
Hasil dalam Tabel 6.1 dan 6.2 didapat dengan parameter pengurangan *learning rate* 0,01, dan pasangan data latih dan uji dengan perbandingan 50:50 yang dibatasi dengan maksimum *epoch* 10.000. Percobaan dilakukan sebanyak sepuluh

kali agar mendapatkan hasil akurasi yang dapat mewakili kemampuan metode LVQ secara utuh. Untuk mempermudah pengambilan informasi dari tabel tersebut dibuat Grafik Gambar 6.3 dan 6.4.



Gambar 6.3 Grafik peningkatan *learning rate* dengan peningkatan 0,01

Dari Gambar 6.3 menunjukkan untuk setiap kenaikan 0,01 dengan range *learning rate* 0,01 sampai 0,1 nilai akurasi cenderung stabil baik untuk pelatihan atau pengujian. Sehingga dapat diambil kesimpulan peningkatan *learning rate* 0,01 tidak terlalu mempengaruhi nilai akurasi secara signifikan.



Gambar 6.4 Grafik peningkatan *learning rate* dengan peningkatan 0,1

Untuk peningkatan *learning rate* 0,1 yang terlihat dari Gambar 6.4 nilai akurasi yang didapat cenderung menurun. Hal tersebut menunjukkan bahwa data PJK memiliki data yang cukup rapat yang besesuaian dengan *scatter plot* dalam analisis data. Sehingga semakin tinggi nilai *learning rate* tidak menjamin akan meningkatkan nilai akurasi.

Dari hasil Percobaan *learning rate* untuk metode LVQ disimpulkan bahwa nilai *learning rate* yang cocok digunakan adalah nilai *learning rate* antara 0,01 sampai 0,1. Nilai *learning rate* yang memperoleh nilai akurasi terbaik untuk proses pengujiannya adalah 0,03. Nilai tersebut akan digunakan dalam proses pengujian selanjutnya.

6.3 Hasil dan Pembahasan Pengujian Parameter Pengurangan *Learning rate* Untuk Metode LVQ

Pengujian ini bertujuan untuk mengetahui hubungan antara tingkat pengurangan *learning rate* terhadap tingkat akurasi yang didapat, Tabel 6.3 pengurangan *learning rate* terhadap akurasi (hasil lengkap dapat dilihat dalam lampiran C.3).

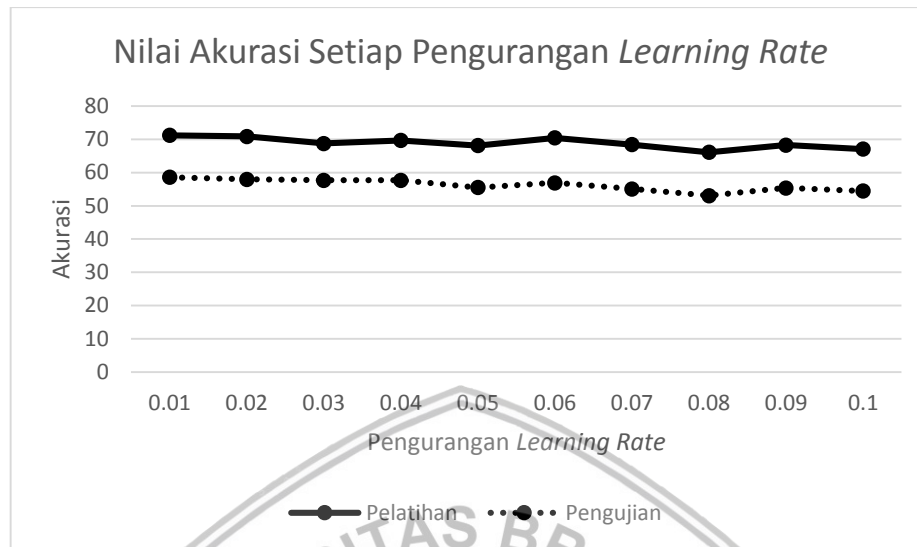
Tabel 6.3 Hasil pengujian parameter pengurangan *learning rate* dalam metode LVQ

| Pengurangan <i>Learning rate</i> | Nilai Akurasi Pelatihan dan Pengujian (%) Percobaan ke -i | | | | Rerata Akurasi | |
|----------------------------------|--|-----------|-----------|-----------|----------------|-----------|
| | 1 | | 10 | | Pelatihan | Pengujian |
| | Pelatihan | Pengujian | Pelatihan | Pengujian | | |
| 0,01 | 73,846 | 58,462 | 72,308 | 58,462 | 71,231 | 58,615 |
| 0,02 | 72,308 | 60,000 | 72,308 | 56,923 | 70,923 | 58,000 |
| 0,03 | 69,231 | 58,462 | 72,308 | 58,462 | 68,769 | 57,692 |
| 0,04 | 67,692 | 60,000 | 70,769 | 58,462 | 69,692 | 57,692 |
| 0,05 | 66,154 | 55,385 | 67,692 | 56,923 | 68,154 | 55,538 |
| 0,06 | 64,615 | 53,846 | 67,692 | 61,538 | 70,462 | 56,923 |
| 0,07 | 63,077 | 55,385 | 67,692 | 61,538 | 68,462 | 55,077 |
| 0,08 | 67,692 | 47,692 | 75,385 | 56,923 | 66,154 | 53,077 |
| 0,09 | 70,769 | 60,000 | 63,077 | 56,923 | 68,308 | 55,385 |
| 0,1 | 66,154 | 53,846 | 61,538 | 61,538 | 67,077 | 54,462 |

Hasil Pengujian Tabel 6.3 didapatkan dengan parameter *learning rate* 0,03 dan pasangan data latih dan uji dengan perbandingan 50:50 yang dibatasi dengan maksimum *epoch* 10.000. Percobaan dilakukan sebanyak sepuluh kali agar mendapatkan hasil akurasi yang dapat mewakili kemampuan metode LVQ secara utuh.

Berdasarkan hasil dalam Tabel 6.3 dibuat grafik Gambar 6.5 yang dapat menunjukkan tren pengaruh setiap parameter pengurangan *learning rate* terhadap nilai akurasi. Dari Gambar 6.5 terlihat jika semakin tinggi nilai pengurangan *learning rate*-nya, nilai akurasi cenderung menurun sehingga dapat disimpulkan nilai parameter pengurangan *learning rate* yang cocok digunakan untuk data PJK adalah nilai parameter pengurangan *learning rate* dengan nilai

antara 0,01 sampai 0,04. Nilai parameter pengurangan *learning rate* yang menghasilkan nilai akurasi terbaik adalah 0,01.



Gambar 6.5 Grafik pengurangan *learning rate* (0.01 sampai 0.1)

6.4 Hasil dan Pembahasan Pengujian Parameter *Learning rate* Untuk Metode *Backpropagation*

Pengujian ini bertujuan untuk mengetahui hubungan antara tingkat *learning rate* terhadap tingkat akurasi yang didapat, Tabel 6.4 dan 6.5 merupakan hasil percobaan tingkat *learning rate* terhadap akurasi (hasil lengkap dapat dilihat dalam lampiran C.2).

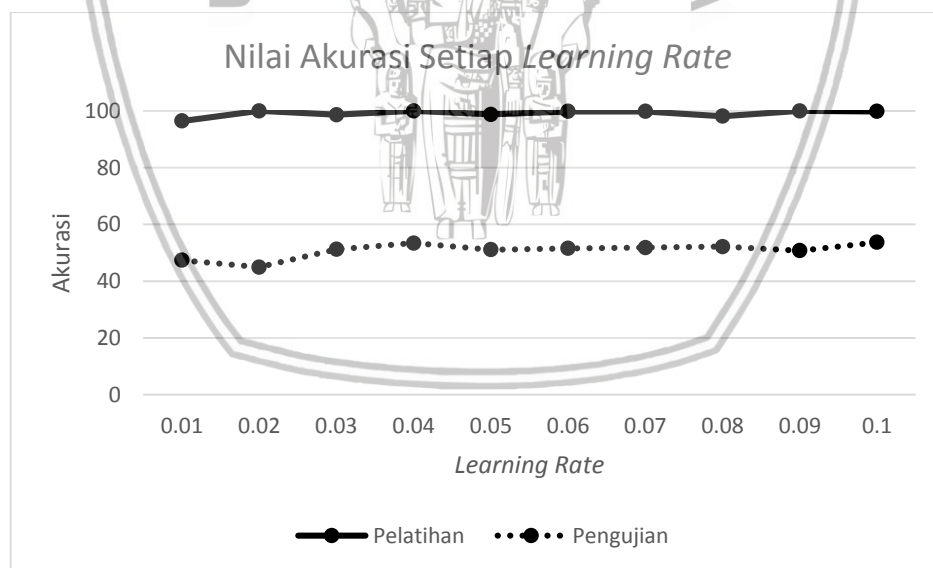
Tabel 6.4 Hasil pengujian parameter *learning rate* dalam metode *Backpropagation* dengan peningkatan 0,01

| <i>Learning rate</i> | Nilai Akurasi Pelatihan dan Pengujian (%) Percobaan ke -i | | | | Rerata Akurasi | |
|----------------------|--|-----------|-----------|-----------|----------------|-----------|
| | 1 | | 10 | | Pelatihan | Pengujian |
| | Pelatihan | Pengujian | Pelatihan | Pengujian | | |
| 0,01 | 100,000 | 49,231 | 81,538 | 41,538 | 96,462 | 47,385 |
| 0,02 | 100,000 | 46,154 | 100,000 | 49,231 | 100,000 | 44,923 |
| 0,03 | 100,000 | 56,923 | 100,000 | 47,692 | 98,615 | 51,231 |
| 0,04 | 100,000 | 53,846 | 100,000 | 56,923 | 100,000 | 53,385 |
| 0,05 | 100,000 | 58,462 | 100,000 | 53,846 | 98,769 | 51,077 |
| 0,06 | 100,000 | 52,308 | 100,000 | 49,231 | 99,846 | 51,538 |
| 0,07 | 100,000 | 52,308 | 100,000 | 53,846 | 99,846 | 51,846 |
| 0,08 | 100,000 | 46,154 | 100,000 | 53,846 | 98,154 | 52,154 |
| 0,09 | 100,000 | 49,231 | 100,000 | 44,615 | 100,000 | 50,769 |
| 0,1 | 100,000 | 58,462 | 98,462 | 52,308 | 99,846 | 53,692 |

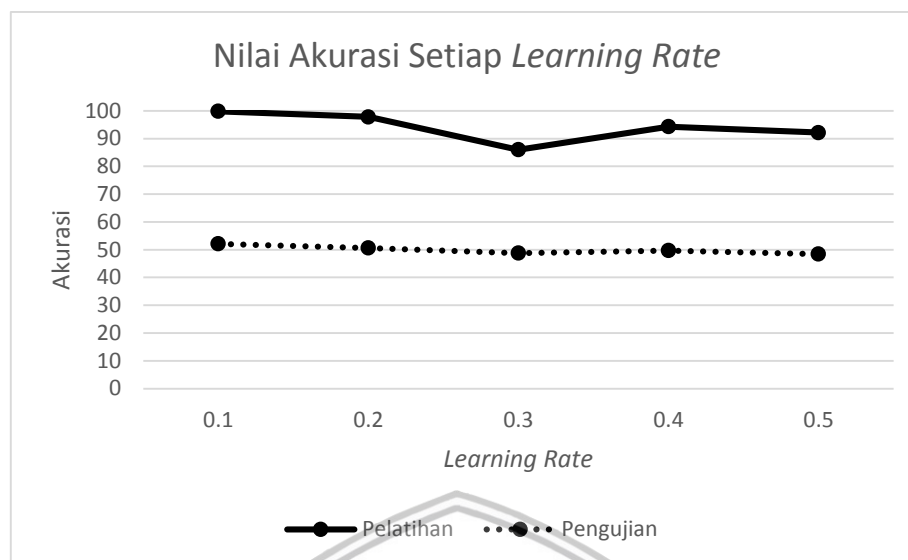
Tabel 6.5 Hasil pengujian parameter *learning rate* dalam metode *Backpropagation* dengan peningkatan 0,1

| Learning rate | Nilai Akurasi Pelatihan dan Pengujian (%) | | | | Rerata Akurasi | |
|---------------|---|-----------|-----------|-----------|----------------|-----------|
| | Percobaan ke -i | | | | Pelatihan | Pengujian |
| | 1 | | 10 | | | |
| | Pelatihan | Pengujian | Pelatihan | Pengujian | | |
| 0,1 | 100,000 | 49,231 | 100,000 | 61,538 | 99,846 | 52,154 |
| 0,2 | 100,000 | 56,923 | 100,000 | 60,000 | 97,846 | 50,615 |
| 0,3 | 84,615 | 55,385 | 100,000 | 43,077 | 86,000 | 48,769 |
| 0,4 | 76,923 | 50,769 | 93,846 | 53,846 | 94,308 | 49,692 |
| 0,5 | 100,000 | 63,077 | 100,000 | 41,538 | 92,154 | 48,462 |

Hasil dalam Tabel 6.4 dan 6.5 didapat dengan arsitektur dua *hidden layer* yang setiap *layer*-nya memiliki lima neuron, fungsi aktivasi *sigmoid* biner, dan pasangan data latih dan uji dengan perbandingan 50:50 yang dibatasi dengan maksimum *epoch* 10.000. Percobaan dilakukan sebanyak sepuluh kali agar mendapatkan hasil akurasi yang dapat mewakili kemampuan metode *Backpropagation* secara utuh. Untuk mempermudah pengambilan informasi dari tabel tersebut dibuat Grafik Gambar 6.6 dan 6.7.



Gambar 6.6 Grafik peningkatan *learning rate* dengan peningkatan 0,01



Gambar 6.7 Grafik peningkatan *learning rate* dengan peningkatan 0,1

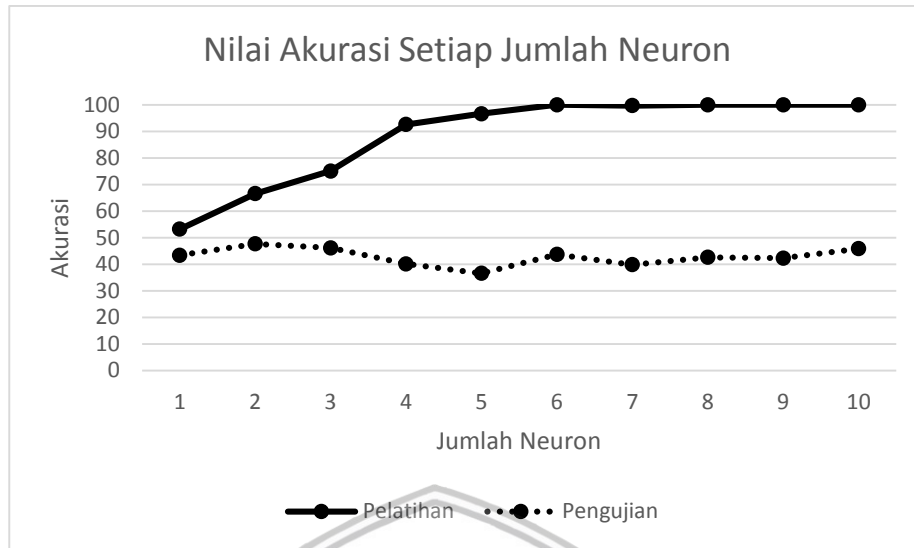
Dari Gambar 6.6 terlihat dengan nilai *learning rate* antara 0,01 sampai 0,1, nilai akurasi cenderung naik. Nilai *learning rate* dengan skala peningkatan 0,1 pada Gambar 6.7, nilai akurasi cenderung sedikit menurun. Dari pengujian ini dapat disimpulkan nilai *learning rate* yang cocok digunakan untuk data PJK adalah 0,1.

6.5 Hasil dan Pembahasan Pengujian Jumlah Neuron dalam Arsitektur *Backpropagation*

Pengujian ini bertujuan untuk mengetahui jumlah neuron yang dapat menghasilkan nilai akurasi yang optimal, Tabel 6.6 merupakan hasil pengujian jumlah neuron(hasil lengkap dapat dilihat dalam lampiran C.4).

Tabel 6.6 Hasil pengujian jumlah neuron dalam metode *Backpropagation*

| Jumlah Neuron | Nilai Akurasi Pelatihan dan Pengujian (%) Percobaan ke -i | | | | Rerata Akurasi | |
|---------------|--|-----------|-----------|-----------|----------------|---------------|
| | 1 | | 10 | | Pelatihan | Pengujian |
| | Pelatihan | Pengujian | Pelatihan | Pengujian | | |
| 1 | 52,308 | 43,077 | 53,846 | 43,077 | 53,231 | 43,385 |
| 2 | 72,308 | 46,154 | 73,846 | 47,692 | 66,615 | 47,692 |
| 3 | 61,538 | 47,692 | 83,077 | 46,154 | 75,077 | 46,154 |
| 4 | 80,000 | 30,769 | 96,923 | 44,615 | 92,615 | 40,154 |
| 5 | 100,000 | 41,538 | 100,000 | 40,000 | 96,615 | 36,615 |
| 6 | 100,000 | 41,538 | 100,000 | 41,538 | 100,000 | 43,692 |
| 7 | 100,000 | 40,000 | 100,000 | 36,923 | 99,692 | 39,846 |
| 8 | 100,000 | 40,000 | 100,000 | 49,231 | 100,000 | 42,615 |
| 9 | 100,000 | 47,692 | 100,000 | 46,154 | 100,000 | 42,308 |
| 10 | 100,000 | 53,846 | 100,000 | 43,077 | 100,000 | 45,846 |



Gambar 6.8 Grafik peningkatan jumlah neuron

Dari Gambar 6.8 menunjukkan jika semakin banyak jumlah neuron akan meningkatkan nilai akurasi proses pelatihan. Namun, nilai akurasi pengujian cenderung stabil.

6.6 Hasil dan Pembahasan Pengujian Jumlah *Hidden Layer* dalam Arsitektur *Backpropagation*

Pengujian ini bertujuan untuk mengetahui jumlah *Hidden Layer* yang dapat menghasilkan nilai akurasi yang optimal dan lama waktu proses pelatihannya. Tabel 6. merupakan hasil pengujian jumlah *hidden layer* (hasil lengkap dapat dilihat dalam lampiran C.5).

Tabel 6.7 Hasil pengujian jumlah *hidden layer* dalam metode *Backpropagation*

| Jumlah <i>Hidden Layer</i> | Nilai Akurasi Pelatihan dan Pengujian (%) Percobaan ke -i | | | | Rerata Akurasi | |
|----------------------------|--|-----------|-----------|-----------|----------------|-----------|
| | 1 | | 10 | | Pelatihan | Pengujian |
| | Pelatihan | Pengujian | Pelatihan | Pengujian | | |
| 1 (10) | 100 | 53,84615 | 100 | 43,07692 | 100 | 45,84615 |
| 2 (10 10) | 100 | 53,84615 | 100 | 40 | 100 | 49,38462 |
| 3 (10 10 10) | 100 | 47,69231 | 100 | 55,38462 | 100 | 54,46154 |
| 4 (10 10 10 10) | 100 | 56,92308 | 100 | 41,53846 | 100 | 53,38462 |

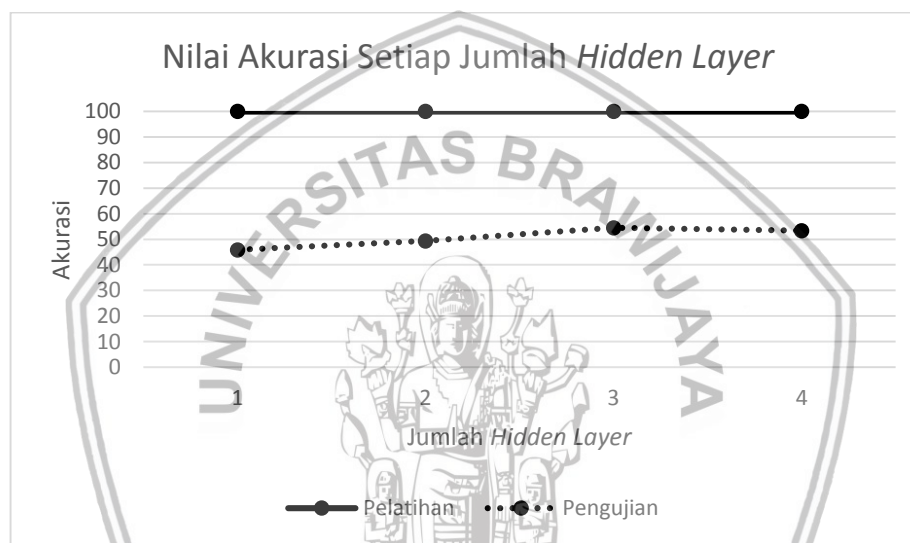
Tabel 6.8 Lama waktu pelatihan setiap jumlah *hidden layer* dalam metode *Backpropagation*

| Jumlah <i>Hidden Layer</i> | Lama Waktu Pelatihan (s) Percobaan ke -i | | Rerata Lama Waktu (s) |
|----------------------------|--|----|-----------------------|
| | 1 | 10 | |

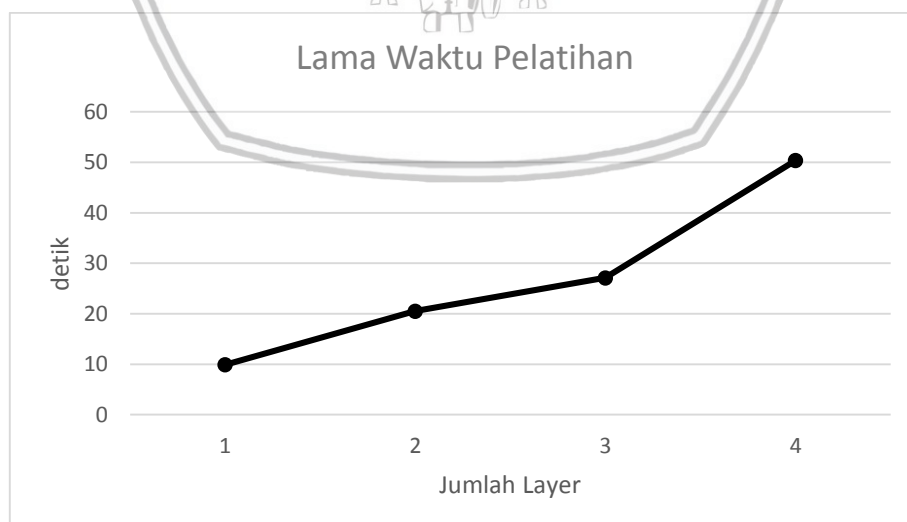
Tabel 6.8 (lanjutan)

| | | | |
|-----------------|--------|--------|---------------|
| 1 (10) | 10,253 | 11,089 | 9,900 |
| 2 (10 10) | 20,135 | 19,648 | 20,529 |
| 3 (10 10 10) | 31,507 | 31,638 | 27,118 |
| 4 (10 10 10 10) | 60,606 | 47,951 | 50,360 |

Berdasarkan Tabel 6.7 dan 6.8 terlihat jika semakin banyak jumlah *hidden layer* maka nilai akurasi pengujian cenderung naik namun tidak signifikan. Waktu pelatihan akan semakin lama jika jumlah *hidden layer* meningkat. Untuk mempermudah gambaran peningkatan akurasi dan lama waktu pelatihan terhadap setiap jumlah *hidden layer* dapat dilihat pada Gambar 6.9 dan 6.10.



Gambar 6.9 Grafik peningkatan jumlah *hidden layer*



Gambar 6.10 Grafik lama waktu pelatihan setiap *hidden layer*

6.7 Hasil dan Pembahasan Pengujian Arsitektur *Backpropagation*

Dari hasil pengujian metode *Backpropagation* sebelumnya, didapatkan nilai akurasi akan semakin baik jika semakin banyak jumlah neuron dan *jumlah hidden layer*nya namun semakin banyak neuron dan *hidden layer* waktu pelatihan akan semakin lama. Berdasarkan kesimpulan tersebut dibuat beberapa jenis arsitektur *Backpropagation* yang akan diuji dengan parameter *learning rate* 0,1, fungsi aktivasi *sigmoid* biner, dan perbandingan pasangan data latih dan uji 50:50, serta *epoch* maksimum 10.000 sehingga didapatkan arsitektur *Backpropagation* yang dapat menghasilkan akurasi optimal dengan waktu pelatihan yang relatif singkat. Hasil pengujiannya dapat dilihat pada Tabel 6.9 dan lama pelatihannya pada Tabel 6.10 (Hasil lengkapnya dapat dilihat dalam lampiran C.6).

Tabel 6.9 Hasil pengujian arsitektur *Backpropagation*

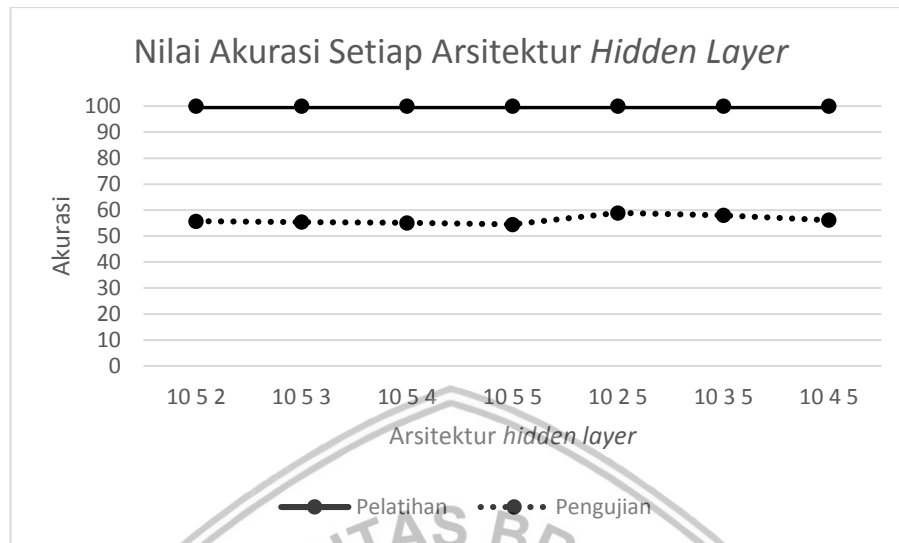
| Arsitektur <i>Hidden Layer</i> | Nilai Akurasi Pelatihan dan Pengujian (%) Percobaan ke -i | | | | Rerata Akurasi | |
|-----------------------------------|--|-----------|-----------|-----------|----------------|-----------|
| | 1 | | 10 | | Pelatihan | Pengujian |
| | Pelatihan | Pengujian | Pelatihan | Pengujian | | |
| 10 5 2 | 100,000 | 56,923 | 100,000 | 56,923 | 100,000 | 55,692 |
| 10 5 3 | 100,000 | 52,308 | 100,000 | 63,077 | 100,000 | 55,385 |
| 10 5 4 | 100,000 | 53,846 | 100,000 | 58,462 | 100,000 | 55,077 |
| 10 5 5 | 100,000 | 56,923 | 100,000 | 49,231 | 100,000 | 54,462 |
| 10 2 5 | 100,000 | 63,077 | 100,000 | 50,769 | 100,000 | 58,923 |
| 10 3 5 | 100,000 | 53,846 | 100,000 | 47,692 | 100,000 | 58,000 |
| 10 4 5 | 100,000 | 60,000 | 100,000 | 52,308 | 100,000 | 56,154 |

Tabel 6.10 Lama waktu pelatihan setiap arsitektur *Backpropagation*

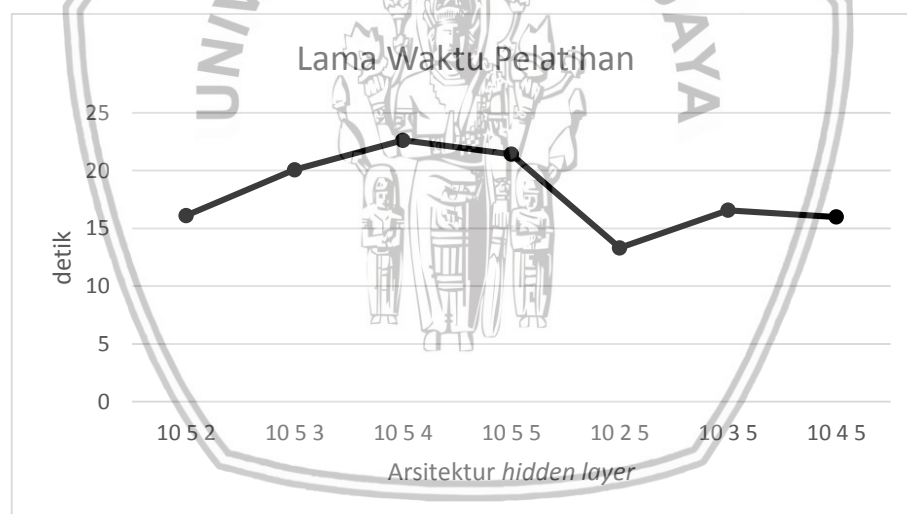
| Arsitektur <i>Hidden Layer</i> | Lama Waktu Pelatihan (s) Percobaan ke -i | | Rerata Lama Waktu (s) |
|-----------------------------------|--|--------|-----------------------------|
| | 1 | 10 | |
| 10 5 2 | 15,314 | 12,619 | 16,104 |
| 10 5 3 | 22,208 | 18,305 | 20,076 |
| 10 5 4 | 24,772 | 18,190 | 22,618 |
| 10 5 5 | 30,658 | 20,829 | 21,415 |
| 10 2 5 | 13,887 | 11,105 | 13,308 |
| 10 3 5 | 16,570 | 17,240 | 16,572 |
| 10 4 5 | 18,135 | 15,204 | 15,988 |

Berdasarkan Tabel 6.9 dan 6.10 dibuat Grafik Gambar 6.11 dan 6.12 agar mempermudah proses analisis. Gambar 6.11 dan 6.12 menunjukkan jika arsitektur (10,2,5), (10,3,5), dan (10,4,5) memiliki nilai akurasi pengujian yang lebih baik dan waktu pelatihan yang relatif singkat sehingga dapat dikatakan bahwa ketiga

arsitektur tersebut merupakan arsitektur yang cocok digunakan untuk pengujian selanjutnya.



Gambar 6.11 Grafik hasil pengujian arsitektur *Backpropagation*



Gambar 6.12 Grafik lama waktu pelatihan setiap arsitektur *Backpropagation*

6.8 Hasil dan Pembahasan Pengujian Jenis Fungsi Aktivasi Untuk Metode *Backpropagation*

Dari pengujian sebelumnya didapatkan arsitektur yang optimal namun fungsi aktivasi yang digunakan hanya *sigmoid* biner. Dalam pengujian ini arsitektur tersebut akan dibandingkan dengan fungsi aktivasi yang berbeda. Hasil pengujian

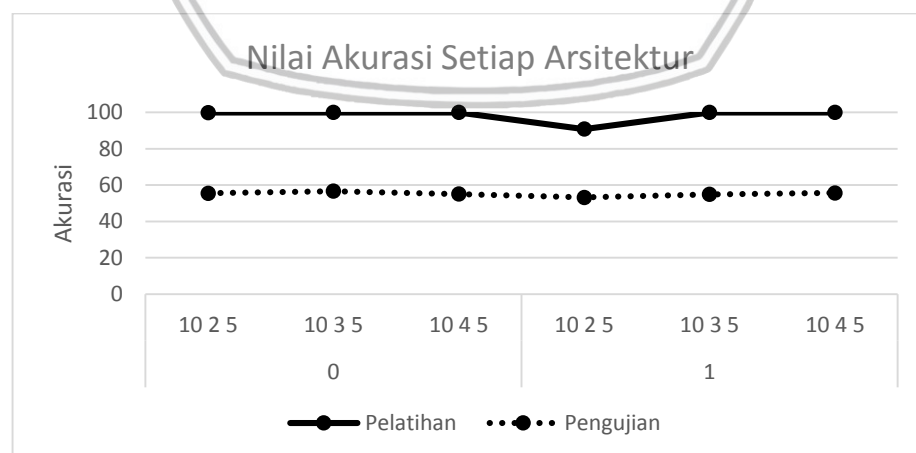
dan lama waktu pelatihannya dapat dilihat dalam Tabel 6.11 dan 6.12. Agar mempermudah proses analisis maka dibuat Grafik Gambar 6.13 dan 6.14.

**Tabel 6.11 Hasil pengujian jenis fungsi aktivasi setiap arsitektur
*Backpropagation***

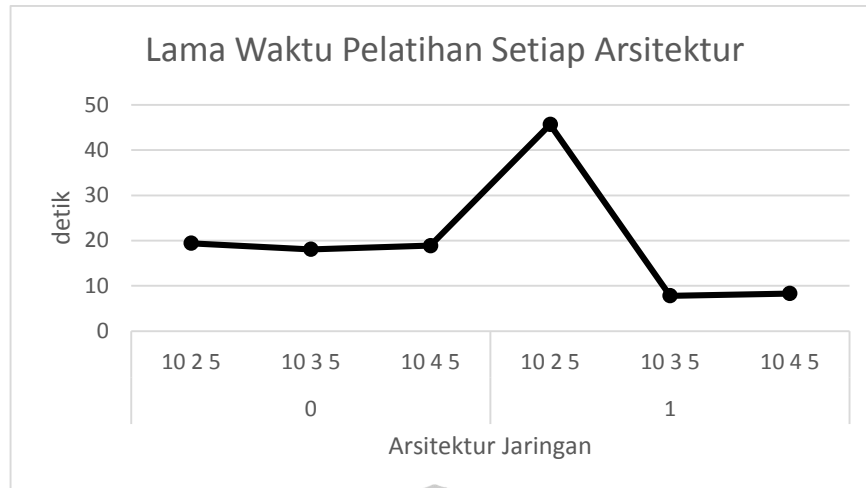
| Jenis Fungsi Aktivasi | Arsitektur Hidden Layer | Nilai Akurasi Pelatihan dan Pengujian (%) Percobaan ke -i | | | | Rerata Akurasi | |
|-----------------------|-------------------------|---|-----------|-----------|-----------|----------------|-----------|
| | | 1 | | 10 | | Pelatihan | Pengujian |
| | | Pelatihan | Pengujian | Pelatihan | Pengujian | | |
| Biner (0) | 10 2 5 | 100,000 | 60,000 | 100,000 | 67,692 | 99,846 | 55,538 |
| | 10 3 5 | 100,000 | 47,692 | 100,000 | 52,308 | 100,000 | 56,615 |
| | 10 4 5 | 100,000 | 50,769 | 100,000 | 53,846 | 100,000 | 55,077 |
| Bipolar (1) | 10 2 5 | 92,308 | 55,385 | 92,308 | 55,385 | 90,769 | 53,231 |
| | 10 3 5 | 100,000 | 44,615 | 100,000 | 60,000 | 100,000 | 54,923 |
| | 10 4 5 | 100,000 | 60,000 | 100,000 | 61,538 | 100,000 | 55,692 |

Tabel 6.12 Lama waktu pelatihan setiap arsitektur *Backpropagation*

| Arsitektur Hidden Layer | Lama Waktu Pelatihan (s) Percobaan ke -i | | Rerata Lama Waktu (s) |
|-------------------------|--|--------|-----------------------|
| | 1 | 10 | |
| 10 2 5 | 14,633 | 14,034 | 19,4246 |
| 10 3 5 | 22,835 | 16,944 | 18,0572 |
| 10 4 5 | 24,005 | 13,915 | 18,8705 |
| 10 2 5 | 91,504 | 54,24 | 45,6513 |
| 10 3 5 | 8,01 | 7,345 | 7,8112 |
| 10 4 5 | 11,878 | 6,23 | 8,3082 |



**Gambar 6.13 Grafik hasil pengujian fungsi aktivasi setiap arsitektur
*Backpropagation***



Gambar 6.14 Grafik lama waktu pelatihan setiap arsitektur *Backpropagation*

Gambar 6.13 menunjukkan jika fungsi aktivasi biner dan bipolar menghasilkan nilai akurasi yang relatif stabil dan sama namun berdasarkan Gambar 6.14 waktu pelatihan dengan fungsi aktivasi *sigmoid* bipolar relatif lebih cepat.

Dari pengujian ini didapatkan bahwa fungsi aktivasi biner memiliki tingkat akurasi yang lebih stabil. Jadi, pada penelitian selanjutnya fungsi aktivasi binerlah yang akan digunakan.

6.9 Hasil dan Pembahasan Pasangan Data Latih dan Uji Untuk Metode LVQ dan Backpropagation

Pengujian ini dilakukan untuk membandingkan nilai akurasi yang dihasilkan oleh empat pasangan data latih dan uji yang sudah dibuat. Untuk metode LVQ pasangan data latih dan uji tersebut akan diuji dengan parameter *learning rate* 0,03 dan pengurannya 0,01 lalu untuk metode *Backpropagation* arsitektur yang digunakan (10,3,5) dengan aktivasi *sigmoid* biner dan *learning rate* 0,1. Kedua metode tersebut dibatasi dengan maksimum *epoch* 10.000. Hasil pengujian untuk metode LVQ dapat dilihat pada Tabel 6.13 dan metode *Backpropagation* pada Tabel 6.14.

Tabel 6.13 Hasil pengujian pasangan data latih dan uji untuk metode LVQ

| Pasangan Data Latih dan Uji | Nilai Akurasi Pelatihan dan Pengujian (%) Percobaan ke -i | | | | Rerata Akurasi | |
|-----------------------------|---|-----------|-----------|-----------|----------------|-----------|
| | 1 | | 10 | | Pelatihan | Pengujian |
| | Pelatihan | Pengujian | Pelatihan | Pengujian | | |
| (65,65) / 50:50 | 73,846 | 58,462 | 73,846 | 58,462 | 71,385 | 58,615 |
| (78,52) / 60:40 | 70.513 | 42.308 | 74.359 | 46.154 | 72.949 | 45.000 |

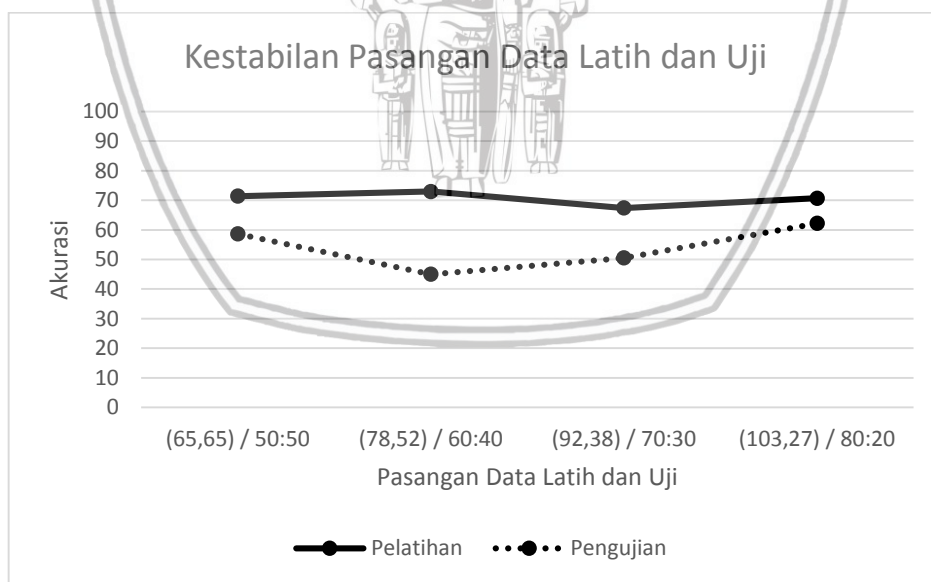
Tabel 6.13 (lanjutan)

| | | | | | | |
|---------------------|--------|--------|--------|--------|---------------|---------------|
| (92,38) / 70:30 | 69,565 | 52,632 | 65,217 | 44,737 | 67,391 | 50,526 |
| (103,27) / 80:20 | 70,874 | 59,259 | 69,903 | 55,556 | 70,680 | 62,222 |

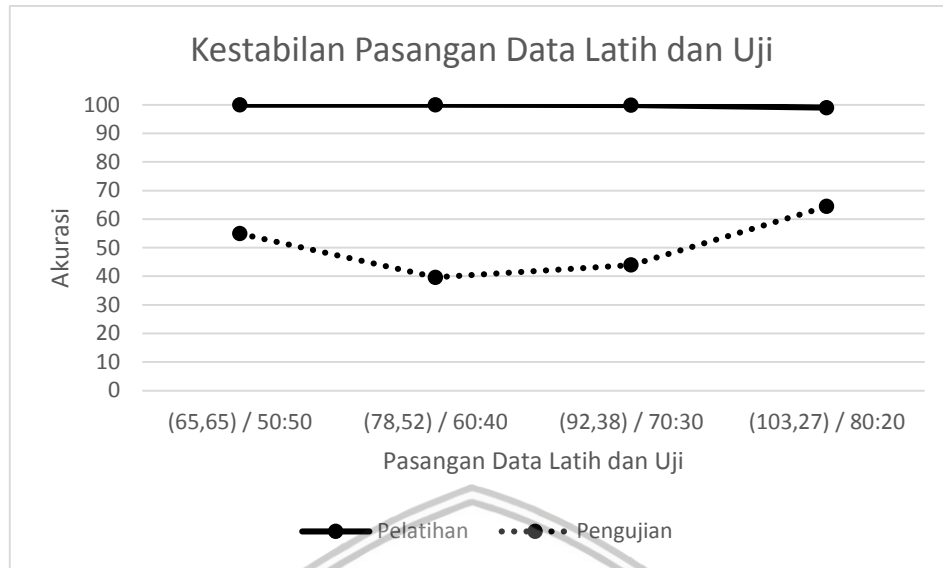
**Tabel 6.14 Hasil pengujian pasangan data latih dan uji untuk metode
Backpropagation**

| Pasangan Data Latih dan Uji | Nilai Akurasi Pelatihan dan Pengujian (%) Percobaan ke -i | | | | Rerata Akurasi | |
|-----------------------------------|--|-----------|-----------|-----------|----------------|---------------|
| | 1 | | 10 | | Pelatihan | Pengujian |
| | Pelatihan | Pengujian | Pelatihan | Pengujian | | |
| (65,65) / 50:50 | 100,000 | 53,846 | 100,000 | 53,846 | 100,000 | 54,923 |
| (78,52) / 60:40 | 100,000 | 38,462 | 100,000 | 40,385 | 100,000 | 39,615 |
| (92,38) / 70:30 | 98,913 | 52,632 | 100,000 | 44,737 | 99,891 | 43,947 |
| (103,27) / 80:20 | 100,000 | 55,556 | 100,000 | 59,259 | 99,029 | 64,444 |

Untuk mempermudah analisis dari Tabel 6.13 dan 6.14 maka dibuat grafik Gambar 6.15 dan Gambar 6.16.



Gambar 6.15 Grafik hasil pengujian pasangan data latih dan uji untuk metode LVQ



Gambar 6.16 Grafik hasil pengujian pasangan data latih dan uji untuk metode *Backpropagation*

Dari Gambar 6.15, metode LVQ menghasilkan nilai akurasi pelatihan dan pengujian yang cukup mirip. Dari Gambar 6.16, metode *Backpropagation* mendapat tingkat akurasi pelatihan yang sangatlah tinggi namun akurasi pengujiannya tidak terlalu tinggi. Hal tersebut mungkin terjadi karena bobot yang dihasilkan pelatihan *Backpropagation* terlalu cocok terhadap data latih, kondisi ini disebut dengan *overfitting* sedangkan dalam metode LVQ akurasi pelatihan tidak terlalu tinggi, artinya pelatihan LVQ mengalami *underfitting*.

Dari pengujian ini, dapat ditarik kesimpulan bahwa perbandingan data latih dan uji yang menghasilkan nilai akurasi yang terbaik adalah pasangan data latih dan uji dengan perbandingan 80:20.

6.10 Hasil dan Pembahasan Pengujian *Cross Validation*

Pengujian ini dilakukan untuk membandingkan kinerja antara metode LVQ dan *Backpropagation* dalam melakukan prediksi dini penyakit jantung koroner. Dalam penelitian ini 130 *dataset* akan dibagi menjadi lima *fold* dengan jumlah data sebanyak 25 sampai 27 *dataset* untuk setiap *fold*-nya. Hal tersebut didasarkan dari hasil perbandingan pasangan data latih dan uji kedua pengujian sebelumnya yang mendapatkan perbandingan data latih dan uji terbaik yaitu 80:20.

Tabel 6.15 *Fold* ke-*i* dan jumlah datanya

| Fold-i | Jumlah Data |
|--------|-------------|
| 1 | 26 |
| 2 | 26 |
| 3 | 26 |

Tabel 6.15 (lanjutan)

| | |
|---|----|
| 4 | 25 |
| 5 | 27 |

Tabel 6.16 Model pengujian *cross validation*

| Fold-i Model-i | 1 | 2 | 3 | 4 | 5 |
|-------------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| 1 (104,26) | Data Testing | Data Traning | Data Traning | Data Traning | Data Traning |
| 2 (104,26) | Data Traning | Data Testing | Data Traning | Data Traning | Data Traning |
| 3 (104,26) | Data Traning | Data Traning | Data Testing | Data Traning | Data Traning |
| 4 (105,25) | Data Traning | Data Traning | Data Traning | Data Testing | Data Traning |
| 5 (103,27) | Data Traning | Data Traning | Data Traning | Data Traning | Data Testing |

Tabel 6.17 Hasil pengujian *cross validation* metode LVQ

| Pasangan Data Latih dan Uji | Nilai Akurasi Pelatihan dan Pengujian (%) Percobaan ke -i | | | | Rerata Akurasi | |
|--------------------------------------|--|-----------|-----------|-----------|----------------|---------------|
| | 1 | | 10 | | Pelatihan | Pengujian |
| | Pelatihan | Pengujian | Pelatihan | Pengujian | | |
| 1 (104,26) | 67,308 | 50,000 | 67,308 | 50,000 | 65,385 | 51,154 |
| 2 (104,26) | 64,423 | 57,692 | 64,423 | 57,692 | 63,942 | 56,538 |
| 3 (104,26) | 72,115 | 65,385 | 69,231 | 53,846 | 67,308 | 59,231 |
| 4 (105,25) | 68,571 | 52,000 | 66,667 | 52,000 | 65,810 | 52,000 |
| 5 (103,27) | 71,845 | 48,148 | 72,816 | 55,556 | 72,039 | 52,593 |

Tabel 6.18 Lama waktu pelatihan pengujian *cross validation* metode LVQ

| Pasangan Data Latih dan Uji | Lama Waktu Pelatihan (s) Percobaan ke -i | | Rerata Lama Waktu (s) |
|-----------------------------------|--|--------|-----------------------------|
| | 1 | 10 | |
| 1 (104,26) | 18,088 | 17,518 | 17,3612 |
| 2 (104,26) | 18,325 | 17,553 | 17,6091 |
| 3 (104,26) | 18,106 | 17,273 | 17,4849 |

Tabel 6.18 (lanjutan)

| | | | |
|------------|--------|--------|----------------|
| 4 (105,25) | 18,835 | 17,574 | 17,6098 |
| 5 (103,27) | 18,475 | 17,411 | 17,4072 |

Tabel 6.17 merupakan hasil pengujian *cross validation* metode LVQ. Parameter LVQ yang digunakan dalam pengujian tersebut adalah *learning rate* 0,03, pengurangan *learning rate* 0,01, dan *epoch* maksimum 10.000. Parameter tersebut didapatkan dari hasil pengujian metode LVQ sebelumnya.

Dari Tabel 6.17 menunjukkan bahwa proses pelatihan metode LVQ tidak bisa mencapai konvergen (nilai akurasi dalam pelatihan rendah). Hal tersebut disebabkan data penyakit koroner yang bersifat menyebar. Sifat data penyakit jantung koroner tersebut juga menyebabkan akurasi pengujian metode LVQ juga menjadi buruk. Dari Tabel 6.18 menunjukkan bahwa waktu pelatihan metode LVQ cukup singkat.

Tabel 6.19 Hasil pengujian *cross validation* metode *Backpropagation*

| Pasangan Data Latih dan Uji | Nilai Akurasi Pelatihan dan Pengujian (%) | | | | Rerata Akurasi | |
|-----------------------------|---|--------|---------|--------|----------------|---------------|
| | Percobaan ke -i | | | | Pelatihan | Pengujian |
| | 1 | 10 | 1 | 10 | | |
| 1 (104,26) | 100,000 | 57,692 | 89,423 | 69,231 | 93,462 | 67,692 |
| 2 (104,26) | 88,462 | 76,923 | 100,000 | 69,231 | 97,788 | 73,077 |
| 3 (104,26) | 97,115 | 69,231 | 96,154 | 69,231 | 97,019 | 73,077 |
| 4 (105,25) | 98,095 | 84,000 | 96,190 | 72,000 | 98,190 | 74,400 |
| 5 (103,27) | 100,000 | 51,852 | 100,000 | 51,852 | 93,495 | 55,556 |

Tabel 6.20 Lama waktu pelatihan pengujian *cross validation* metode *Backpropagation*

| Pasangan Data Latih dan Uji | Lama Waktu Pelatihan (s) | | Rerata Lama Waktu (s) |
|-----------------------------|--------------------------|--------|-----------------------|
| | 1 | 10 | |
| 1 (104,26) | 54,611 | 93,703 | 65,0054 |
| 2 (104,26) | 98,769 | 27,601 | 69,2143 |
| 3 (104,26) | 97,454 | 90,653 | 73,6203 |
| 4 (105,25) | 98,592 | 93,813 | 94,0894 |
| 5 (103,27) | 26,216 | 33,176 | 48,9022 |

Tabel 6.19 merupakan hasil pengujian *cross validation* metode *Backpropagation*. Parameter *Backpropagation* yang digunakan dalam pengujian tersebut adalah fungsi aktivasi *sigmoid* biner, *learning rate* 0,1, jumlah *hidden layer* 3 dan pasangan neuronnya (10,3,5), dan *epoch* maksimum 10.000. Parameter tersebut didapatkan dari hasil pengujian metode *Backpropagation*.

Dari Tabel 6.19 menunjukkan bahwa proses pelatihan metode *Backpropagation* bisa mencapai konvergen (nilai akurasi dalam pelatihan yang tinggi) walaupun data tersebut bersifat menyebar. Hal tersebut menunjukkan toleransi metode *Backpropagation* yang baik terhadap data yang bersifat menyebar. Namun, sifat data penyakit jantung koroner tersebut menyebabkan akurasi pengujian metode *Backpropagation* menjadi buruk. Dari Tabel 20 menunjukkan proses pelatihan *Backpropagation* relatif lama.

Tabel 6.21 Rerata akurasi data latih dan uji metode LVQ dan *Backpropagation*

| Metode | Rerata Akurasi Pelatihan (%) | Rerata Akurasi Pengujian (%) | Rerata Lama Pelatihan (s) |
|------------------------|------------------------------|------------------------------|---------------------------|
| LVQ | 66,89659 | 54,30313 | 17,49444 |
| <i>Backpropagation</i> | 95,99097 | 68,76034 | 70,16632 |

Dari hasil dalam Tabel 6.21 dapat diambil kesimpulan jika metode yang lebih cocok antara LVQ dengan *Backpropagation* dalam melakukan prediksi dini penyakit jantung koroner dengan sepuluh parameter dan empat jenis klasifikasi adalah metode *Backpropagation* dengan perbandingan rerata akurasi pelatihan antara *Backpropagation* dengan LVQ sebesar 95,99097%: 66,89659% dan rerata akurasi pengujian sebesar 68,76034%: 54,30313%. Dari segi waktu pelatihan, metode LVQ lebih cepat dari pada metode *Backpropagation* karena arsitektur LVQ yang lebih sederhana, yakni hanya terdiri dari *layer input* dan *output*.

BAB 7 PENUTUP

Bab ini berisi kesimpulan dari penelitian ini yang dapat menjawab rumusan masalah yang telah dibuat dan saran untuk meningkatkan mutu penelitian selanjutnya.

7.1 Kesimpulan

Berdasarkan hasil pengujian dan analisis dalam penelitian ini yakni perbandingan jaringan saraf tiruan LVQ dan *Backpropagation* dalam deteksi dini penyakit jantung koroner dapat diambil kesimpulan sebagai berikut.

1. Parameter yang digunakan agar mendapat hasil optimal dalam deteksi dini penyakit jantung koroner adalah:
 - a Untuk LVQ, digunakan parameter *learning rate* sebesar 0,03 dan pengurangan *learning rate*-nya 0,01 dengan maksimum *epoch* 10.000.
 - b Untuk *Backpropagation*, digunakan parameter fungsi aktivasi biner; *learning rate* sebesar 0,1; jumlah *hidden layer* sebanyak 3 dengan jumlah neuron setiap *hidden layer*-nya berjumlah 10, 3, dan 5; dan dibatasi dengan maksimum *epoch* 10.000.
2. Perbandingan hasil akurasi deteksi penyakit jantung koroner antara LVQ dengan *Backpropagation* adalah 66,89659% untuk LVQ berbanding 95,99097% untuk *Backpropagation* saat pengujian dengan data latih dan 54,30313% untuk LVQ berbanding 68,76034% untuk *Backpropagation* saat pengujian dengan data uji.
3. Berdasarkan nilai akurasi pengujian data latih dan data ujinya maka dalam penelitian ini metode *Backpropagation* lebih efektif daripada LVQ untuk deteksi dini penyakit jantung koroner.

7.2 Saran

Saran yang dapat digunakan dalam penelitian selanjutnya adalah sebagai berikut.

1. Menggunakan kombinasi dan jumlah variabel penyakit jantung koroner yang berbeda karena dalam penelitian ini dengan kombinasi sepuluh variabel yang digunakan diperoleh hasil yang kurang optimal.
2. Menggunakan variasi metode LVQ yang lain, seperti LVQ2, LVQ2.1, GLVQ dan menggunakan variasi metode *Backpropagation*, seperti *Backpropagation* dengan *adaptive learning rate*, dengan momentum, gabungan keduanya, atau *Backpropagation Levenberg-Marquardt*.

DAFTAR PUSTAKA

- Aladokter.com, 2017a. *Diagnosis Penyakit Jantung - Alodokter*. [daring] Tersedia pada: <<http://www.alodokter.com/penyakit-jantung/diagnosis>> [Diakses 7 Sep 2017].
- Aladokter.com, 2017b. *Penyakit Jantung - jenis, gejala dan penyebab - Alodokter*. [daring] Tersedia pada: <<http://www.alodokter.com/penyakit-jantung>> [Diakses 7 Sep 2017].
- Atthina, N. dan Iswari, L., 2014. Klasterisasi Data Kesehatan Penduduk untuk Menentukan Rentang Derajat Kesehatan Daerah dengan Metode K-Means. *Seminar Nasional Aplikasi Teknologi Infromasi (SNATI)*, [daring] 1, hal.B-52-B-59. Tersedia pada: <<http://journal.uui.ac.id/index.php/Snati/article/view/3246>>.
- Effendy, N. dan Faisal, A., 2008. Prediksi Penyakit Jantung Koroner (Pjk) Berdasarkan Faktor Risiko Menggunakan Jaringan Syaraf Tiruan Backpropagation. *Seminar Nasional Aplikasi Teknologi Informasi (SNATI)*, 1(Snati), hal.1.
- Fausett, L. V, 1994. *Fundamentals of neural networks*. Prentice-Hall.
- Hamidi, R., Furqon, M.T. dan Rahayudi, B., 2017. Implementasi Learning Vector Quantization (LVQ) untuk Klasifikasi Kualitas Air Sungai. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, [daring] 1(12), hal.1758–1763. Tersedia pada: <<http://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/635>>.
- Hawickhorst, B.A. dan Zahorian, S.A., 1996. A Comparison of Three Neural Network Architectures for Automatic Speech Recognition. *International Geoscience and Remote Sensing Symposium IGARSS'96 USA*, 1, hal.787–789.
- Hyndman, R.J. dan Koehler, A.B., 2006. Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22(4), hal.679–688.
- Kayadoe, M., Rumlawang, F.Y. dan Lesnussa, Y.A., 2013. Sistem pengenalan barcode menggunakan jaringan syaraf tiruan backpropagation. *Prosiding FMIPA Universitas Pattimura 2013*, [daring] 2(1), hal.206–213. Tersedia pada: <https://ejournal.unpatti.ac.id/ppr_paperinfo_Ink.php?id=522>.
- Kertohoesodo, S., 1982. *Memelihara Jantung Sehat dan Menjaga Jantung Sakit*. Jakarta: Citra Budaya & CV. Karya Pembina Bangsa.
- Kusumadewi, S., 2003. *Artificial Intelligence (Teknik dan Aplikasinya)*. Yogyakarta: Graha Ilmu.
- Kusumadewi, S., 2004. *Membangun Jaringan Syaraf Tiruan (Menggunakan Matlab dan Excel Link)*. Yogyakarta: Graha Ilmu.
- Leleury, Z.A., Lesnussa, Y.A. dan Madiuw, J., 2016. Sistem Diagnosa Penyakit Dalam dengan Menggunakan Jaringan Saraf Tiruan Metode Backpropagation dan

- Learning Vector Quantization. *Jurnal Matematika Integratif*, [daring] 12(2), hal.89–98. Tersedia pada: <<http://jurnal.unpad.ac.id/jmi/article/view/11925/5553>>.
- Luthfie, S.N., 2012. Implementasi Jaringan Saraf Tiruan Backpropagation Pada Aplikasi Pengenalan Wajah Dengan Jarak Yang Berbeda Menggunakan MATLAB 7.0. *Skripsi Program Studi Teknik Informatika*. [daring] Tersedia pada: <http://www.gunadarma.ac.id/library/articles/graduate/computer-science/2008/Artikel_50403683.pdf>.
- Megawati dan Candra, R.M., 2017. Diagnosa Hama dan Penyakit pada Tanaman Jeruk dengan Menerapkan Jaringan Syaraf Tiruan Learning Vector Quantization (Studi Kasus : Badan Penyuluhan Pertanian Kuok). *Jurnal CoreIT*, 3(2), hal.59–62.
- Putra, J.W.G., 2018. Pengenalan Konsep Pembelajaran Mesin dan Deep Learning. [daring] Tersedia pada: <https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=5&ved=2ahUKEwi35pea2pnfAhXLdH0KHfWeBNwQFjAEegQIAxAC&url=https://www.researchgate.net/profile/Jan_Wira_Gotama_Putra/publication/323700644_Pengenalan_Pembelajaran_Mesin_dan_Deep_Learning/links/5b45c3940f7e9b1c72238a80/Pengenalan-Pembelajaran-Mesin-dan-Deep-Learning.pdf?origin=publication_detail&usg=AOvVaw1a-Z19iCRZnoQg3DLPV6Y7>.
- Raharjo, B., 2011. *Presisi dan Akurasi* «raharjo.org. [daring] Tersedia pada: <<https://beniraharjo.wordpress.com/2011/03/17/presisi-dan-akurasi/>> [Diakses 27 Apr 2018].
- Refaeilzadeh, P., Tang, L. dan Liu, H., 2008. *Cross-Validation*. [daring] Tersedia pada: <<http://leitang.net/papers/ency-cross-validation.pdf>>.
- Utami, A.T.W. dan Ulama, B.S.S., 2015. Penerapan Backpropagation untuk Meningkatkan Efektivitas Waktu dan Akurasi pada Data Wall-Following Robot Navigation. *Jurnal Sains dan Seni ITS*, 4(2).
- Wardhani, R.S., 2014. *Aplikasi Sistem Fuzzy untuk Diagnosa Penyakit Jantung Koroner (Coronary Heart Disease)*. [daring] UNY. Tersedia pada: <<http://eprints.uny.ac.id/12689/>> [Diakses 7 Sep 2017].
- Widodo, T.S., 2013. *Sistem Neuro Fuzzy untuk Pengolahan Informasi, Pemodelan, dan Kendali*. Pertama ed. Yogyakarta: Graha Ilmu.

LAMPIRAN A DATA 130 PENYAKIT JANTUNG KORONER

| No | Jenis Kelamin | Usia | Denyut Nadi | Tekanan Darah Sistolic | Kolestrol | Gula Darah | Trigliserida | Nyeri Dada | Sesak Nafas | Batuk | Diagnosa |
|----|---------------|------|-------------|------------------------|-----------|------------|--------------|------------|-------------|-------|----------|
| 1 | p | 68 | 81 | 140 | 140 | 255 | 143 | Tidak | Ya | Ya | PJKtipe2 |
| 2 | p | 73 | 89 | 120 | 143 | 560 | 190 | Tidak | Ya | Ya | PJKtipe1 |
| 3 | p | 89 | 93 | 117 | 135 | 214 | 289 | Ya | Ya | Ya | PJKtipe2 |
| 4 | p | 70 | 100 | 140 | 164 | 400 | 267 | Tidak | Tidak | Tidak | PJKtipe1 |
| 5 | p | 64 | 98 | 180 | 162 | 176 | 309 | Ya | Tidak | Tidak | PJKtipe2 |
| 6 | p | 70 | 116 | 230 | 215 | 157 | 129 | Ya | Ya | Tidak | PJKtipe3 |
| 7 | l | 66 | 89 | 150 | 154 | 267 | 122 | Ya | Ya | Tidak | PJKtipe3 |
| 8 | l | 90 | 80 | 136 | 144 | 156 | 222 | Ya | Tidak | Tidak | PJKtipe2 |
| 9 | p | 74 | 100 | 148 | 135 | 315 | 370 | Ya | Ya | Tidak | PJKtipe3 |
| 10 | p | 59 | 100 | 120 | 252 | 487 | 241 | Tidak | Tidak | Ya | PJKtipe1 |
| 11 | l | 61 | 90 | 150 | 89 | 500 | 67 | Ya | Ya | Tidak | PJKtipe2 |
| 12 | l | 64 | 100 | 130 | 105 | 208 | 240 | Ya | Ya | Tidak | PJKtipe2 |
| 13 | p | 40 | 90 | 120 | 139 | 71 | 110 | Ya | Ya | Tidak | PJKtipe2 |
| 14 | p | 54 | 100 | 140 | 300 | 212 | 70 | Tidak | Ya | Ya | PJKtipe1 |
| 15 | l | 64 | 90 | 190 | 162 | 262 | 60 | Ya | Ya | Tidak | PJKtipe2 |
| 16 | l | 58 | 105 | 163 | 82 | 163 | 102 | Tidak | Ya | Tidak | PJKtipe2 |
| 17 | p | 62 | 89 | 140 | 184 | 570 | 100 | Ya | Ya | Ya | PJKtipe2 |
| 18 | p | 60 | 100 | 139 | 400 | 490 | 231 | Ya | Tidak | Tidak | PJKtipe1 |
| 19 | l | 70 | 99 | 130 | 171 | 222 | 85 | Ya | Ya | Ya | PJKtipe2 |
| 20 | p | 62 | 100 | 110 | 200 | 540 | 223 | Tidak | Ya | Tidak | PJKtipe1 |
| 21 | l | 76 | 70 | 150 | 386 | 125 | 450 | Ya | Ya | Tidak | PJKtipe3 |
| 22 | l | 71 | 70 | 140 | 256 | 145 | 200 | Tidak | Tidak | Tidak | PJKtipe1 |

| | | | | | | | | | | | |
|----|---|----|-----|-----|-----|-----|-----|-------|-------|-------|----------|
| 23 | l | 43 | 106 | 139 | 232 | 178 | 100 | Ya | Ya | Ya | PJKtipe3 |
| 24 | p | 78 | 95 | 144 | 232 | 343 | 290 | Ya | Ya | Ya | PJKtipe3 |
| 25 | p | 78 | 100 | 142 | 200 | 178 | 310 | Ya | Ya | Ya | PJKtipe3 |
| 26 | l | 52 | 60 | 170 | 231 | 341 | 520 | Ya | Ya | Tidak | PJKtipe3 |
| 27 | l | 52 | 80 | 110 | 245 | 478 | 120 | Ya | Ya | Tidak | PJKtipe2 |
| 28 | l | 49 | 102 | 180 | 198 | 470 | 100 | Ya | Ya | Ya | PJKtipe2 |
| 29 | l | 90 | 75 | 140 | 94 | 143 | 190 | Ya | Ya | Ya | PJKtipe1 |
| 30 | l | 62 | 80 | 185 | 129 | 116 | 252 | Ya | Ya | Tidak | PJKtipe3 |
| 31 | l | 78 | 110 | 154 | 148 | 223 | 67 | Ya | Ya | Tidak | PJKtipe2 |
| 32 | p | 66 | 80 | 120 | 200 | 578 | 97 | Ya | Ya | Tidak | PJKtipe2 |
| 33 | p | 73 | 80 | 219 | 143 | 120 | 157 | Tidak | Ya | Tidak | PJKtipe1 |
| 34 | l | 45 | 96 | 137 | 350 | 327 | 480 | Ya | Ya | Tidak | PJKtipe3 |
| 35 | p | 57 | 90 | 112 | 230 | 80 | 80 | Ya | Ya | Tidak | PJKtipe1 |
| 36 | p | 49 | 112 | 120 | 212 | 150 | 85 | Ya | Tidak | Ya | PJKtipe2 |
| 37 | l | 67 | 85 | 110 | 300 | 343 | 310 | Ya | Ya | Tidak | PJKtipe3 |
| 38 | p | 82 | 93 | 155 | 267 | 164 | 121 | Ya | Ya | Ya | PJKtipe2 |
| 39 | p | 72 | 123 | 184 | 93 | 375 | 112 | Ya | Ya | Tidak | PJKtipe2 |
| 40 | l | 76 | 90 | 130 | 174 | 81 | 60 | Ya | Ya | Tidak | PJKtipe2 |
| 41 | p | 72 | 120 | 150 | 81 | 55 | 57 | Ya | Ya | Ya | PJKtipe2 |
| 42 | l | 58 | 62 | 129 | 110 | 124 | 55 | Tidak | Ya | Ya | PJKtipe1 |
| 43 | l | 62 | 98 | 140 | 150 | 143 | 86 | Tidak | Ya | Ya | PJKtipe1 |
| 44 | p | 73 | 102 | 180 | 169 | 456 | 104 | Ya | Ya | Ya | PJKtipe2 |
| 45 | p | 69 | 98 | 140 | 367 | 160 | 90 | Tidak | Ya | Ya | PJKtipe1 |
| 46 | p | 69 | 80 | 130 | 500 | 406 | 90 | Tidak | Tidak | Ya | PJKtipe1 |
| 47 | p | 79 | 80 | 130 | 277 | 143 | 102 | Ya | Ya | Tidak | PJKtipe2 |

| | | | | | | | | | | | |
|----|---|----|-----|-----|-----|-----|-----|-------|-------|-------|----------|
| 48 | l | 50 | 80 | 100 | 125 | 165 | 560 | Ya | Ya | Ya | PJKtipe3 |
| 49 | l | 72 | 90 | 100 | 285 | 126 | 64 | Ya | Ya | Ya | PJKtipe2 |
| 50 | l | 73 | 74 | 140 | 107 | 453 | 580 | Ya | Ya | Ya | PJKtipe3 |
| 51 | l | 71 | 80 | 140 | 210 | 223 | 112 | Tidak | Ya | Ya | PJKtipe1 |
| 52 | p | 42 | 86 | 100 | 278 | 245 | 113 | Tidak | Ya | Ya | PJKtipe1 |
| 53 | p | 62 | 89 | 170 | 180 | 245 | 50 | Ya | Ya | Ya | PJKtipe1 |
| 54 | p | 78 | 95 | 170 | 477 | 534 | 102 | Ya | Ya | Tidak | PJKtipe3 |
| 55 | l | 74 | 96 | 129 | 386 | 170 | 66 | Tidak | Ya | Tidak | PJKtipe2 |
| 56 | p | 44 | 90 | 110 | 390 | 90 | 110 | Tidak | Ya | Tidak | PJKtipe2 |
| 57 | p | 59 | 90 | 100 | 96 | 456 | 330 | Ya | Ya | Tidak | PJKtipe3 |
| 58 | l | 74 | 78 | 100 | 342 | 378 | 112 | Tidak | Ya | Tidak | PJKtipe1 |
| 59 | p | 65 | 77 | 133 | 180 | 163 | 80 | Tidak | Ya | Tidak | PJKtipe2 |
| 60 | p | 71 | 92 | 161 | 181 | 167 | 51 | Tidak | Ya | Tidak | PJKtipe2 |
| 61 | p | 65 | 107 | 161 | 181 | 564 | 51 | Tidak | Tidak | Tidak | PJKtipe1 |
| 62 | p | 84 | 58 | 190 | 253 | 341 | 551 | Ya | Ya | Tidak | PJKtipe3 |
| 63 | l | 78 | 83 | 150 | 196 | 112 | 339 | Ya | Ya | Tidak | PJKtipe3 |
| 64 | l | 46 | 120 | 200 | 151 | 381 | 480 | Ya | Ya | Tidak | PJKtipe3 |
| 65 | p | 88 | 91 | 220 | 130 | 254 | 89 | Ya | Ya | Tidak | PJKtipe2 |
| 66 | l | 60 | 83 | 140 | 133 | 154 | 97 | Tidak | Ya | Tidak | PJKtipe1 |
| 67 | l | 57 | 79 | 110 | 125 | 186 | 112 | Ya | Ya | Tidak | PJKtipe2 |
| 68 | p | 45 | 120 | 115 | 300 | 255 | 210 | Ya | Ya | Ya | PJKtipe2 |
| 69 | l | 54 | 52 | 233 | 267 | 307 | 261 | Ya | Tidak | Tidak | PJKtipe2 |
| 70 | l | 76 | 97 | 200 | 390 | 303 | 400 | Ya | Ya | Ya | PJKtipe3 |
| 71 | l | 80 | 95 | 135 | 180 | 85 | 65 | Ya | Ya | Tidak | PJKtipe2 |
| 72 | l | 70 | 80 | 145 | 250 | 145 | 210 | Tidak | Tidak | Tidak | PJKtipe1 |

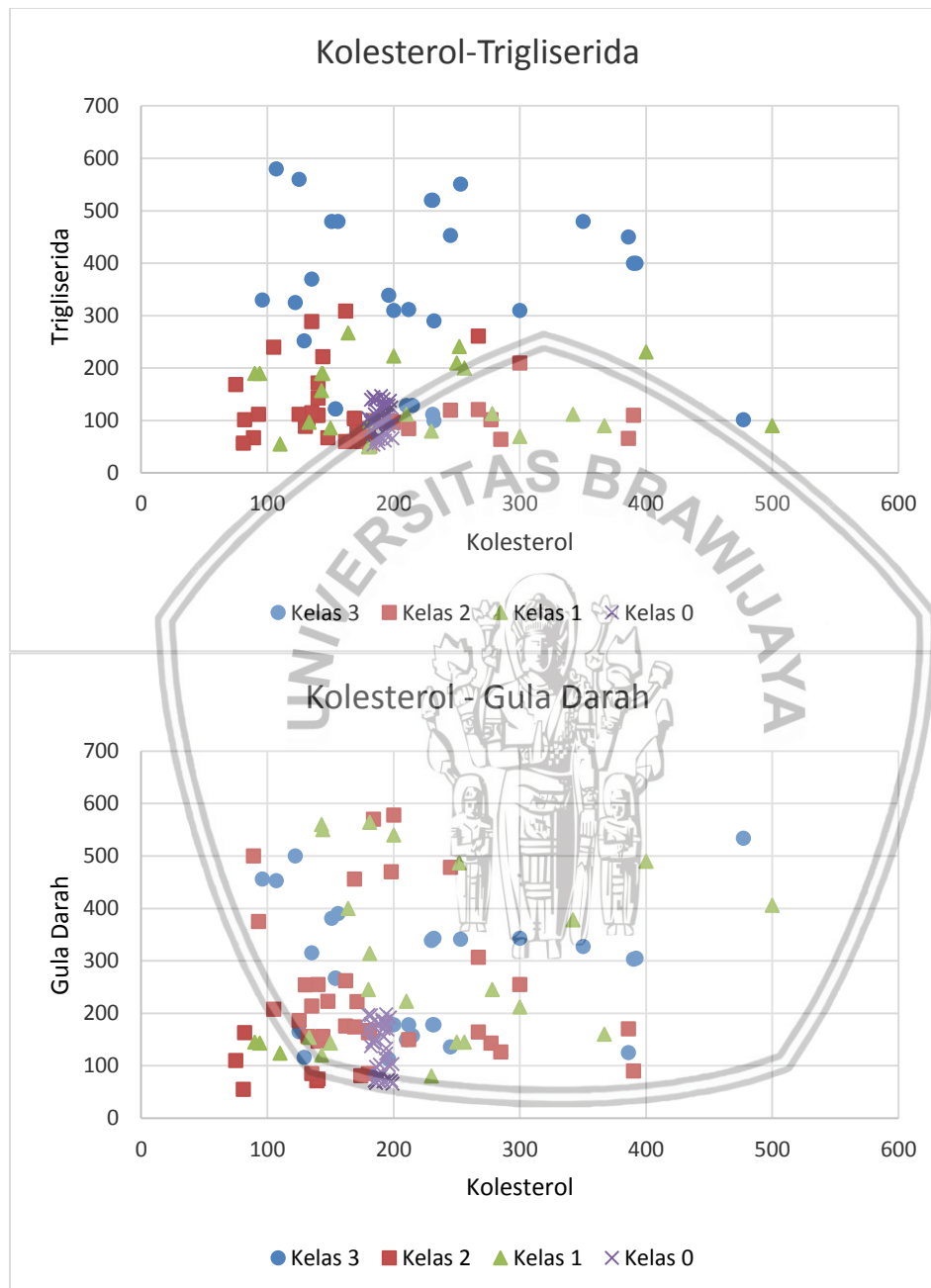
| | | | | | | | | | | | |
|----|---|----|-----|-----|-----|-----|-----|-------|-------|-------|----------|
| 73 | p | 42 | 94 | 119 | 140 | 75 | 110 | Ya | Ya | Tidak | PJKtipe2 |
| 74 | l | 55 | 78 | 90 | 181 | 314 | 99 | Tidak | Ya | Ya | PJKtipe1 |
| 75 | l | 55 | 78 | 142 | 140 | 147 | 172 | Ya | Ya | Tidak | PJKtipe2 |
| 76 | l | 89 | 73 | 144 | 90 | 145 | 190 | Ya | Ya | Ya | PJKtipe1 |
| 77 | p | 74 | 80 | 199 | 132 | 156 | 98 | Tidak | Ya | Ya | PJKtipe2 |
| 78 | l | 83 | 64 | 120 | 135 | 85 | 115 | Tidak | Ya | Tidak | PJKtipe2 |
| 79 | l | 60 | 100 | 140 | 122 | 500 | 325 | Ya | Ya | Ya | PJKtipe3 |
| 80 | p | 72 | 115 | 235 | 210 | 150 | 129 | Ya | Ya | Tidak | PJKtipe3 |
| 81 | p | 70 | 80 | 120 | 144 | 550 | 190 | Tidak | Ya | Ya | PJKtipe1 |
| 82 | p | 73 | 90 | 120 | 169 | 174 | 104 | Tidak | Ya | Ya | PJKtipe2 |
| 83 | l | 45 | 115 | 220 | 156 | 390 | 480 | Ya | Ya | Tidak | PJKtipe3 |
| 84 | l | 60 | 89 | 150 | 154 | 267 | 122 | Ya | Ya | Tidak | PJKtipe3 |
| 85 | p | 80 | 111 | 143 | 212 | 178 | 312 | Ya | Ya | Ya | PJKtipe3 |
| 86 | l | 43 | 106 | 130 | 231 | 178 | 112 | Ya | Ya | Ya | PJKtipe3 |
| 87 | l | 44 | 53 | 164 | 75 | 110 | 169 | Tidak | Tidak | Tidak | PJKtipe2 |
| 88 | l | 50 | 59 | 172 | 230 | 339 | 520 | Ya | Ya | Tidak | PJKtipe3 |
| 89 | l | 55 | 103 | 160 | 245 | 136 | 453 | Ya | Ya | Tidak | PJKtipe3 |
| 90 | l | 75 | 90 | 212 | 392 | 305 | 400 | Ya | Ya | Ya | PJKtipe3 |
| 91 | l | 22 | 76 | 100 | 186 | 98 | 140 | Tidak | Ya | Ya | Normal |
| 92 | l | 23 | 85 | 112 | 183 | 146 | 102 | Tidak | Tidak | Tidak | Normal |
| 93 | p | 23 | 76 | 100 | 194 | 181 | 100 | Tidak | Ya | Tidak | Normal |
| 94 | l | 24 | 62 | 104 | 194 | 123 | 109 | Tidak | Tidak | Tidak | Normal |
| 95 | p | 24 | 89 | 99 | 188 | 178 | 93 | Ya | Ya | Tidak | Normal |
| 96 | l | 27 | 88 | 98 | 182 | 196 | 140 | Ya | Tidak | Tidak | Normal |
| 97 | l | 31 | 64 | 100 | 185 | 150 | 98 | Ya | Ya | Ya | Normal |

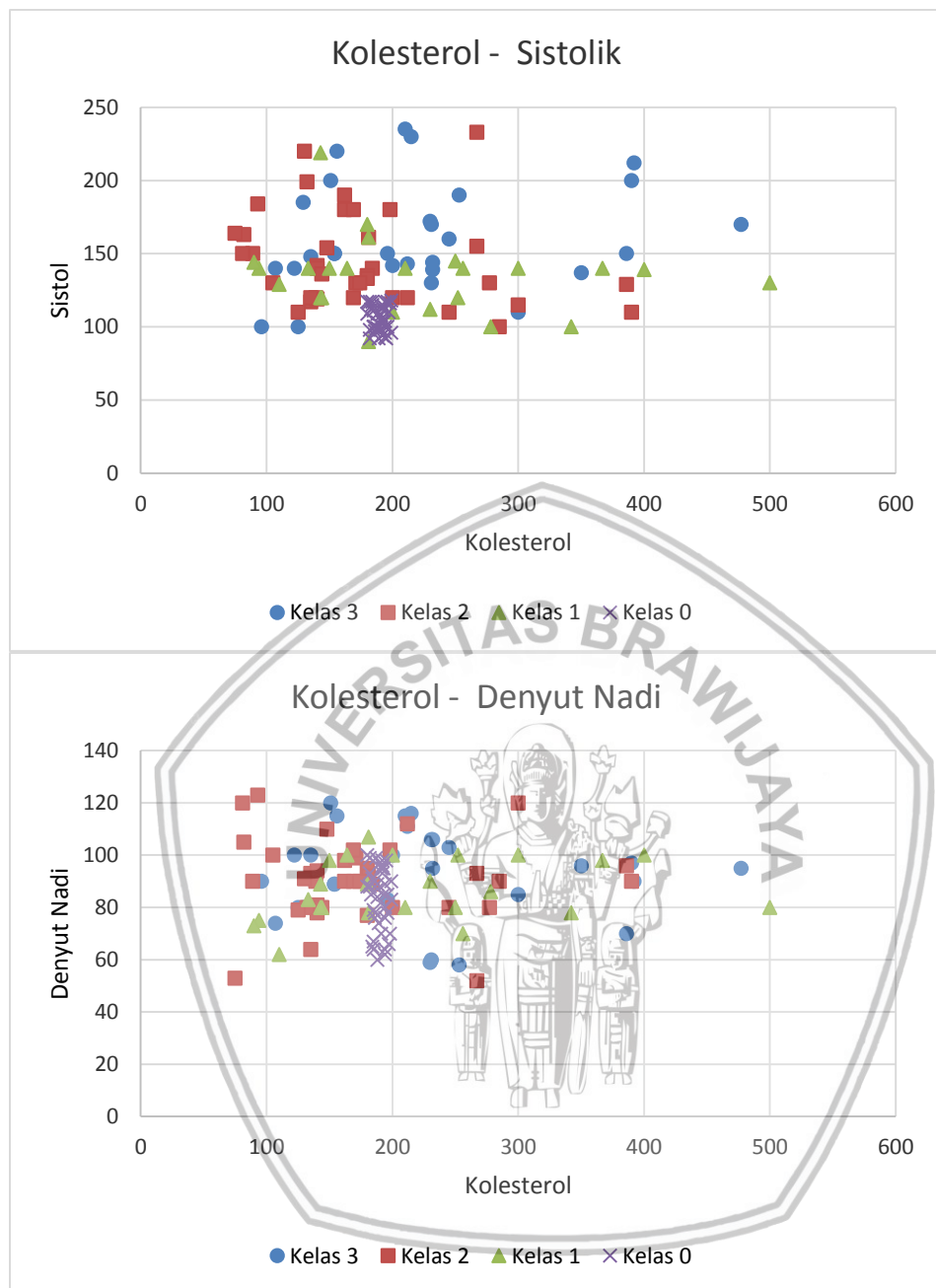
| | | | | | | | | | | | |
|-----|---|----|-----|-----|-----|-----|-----|-------|-------|-------|--------|
| 98 | p | 33 | 100 | 109 | 180 | 169 | 104 | Ya | Ya | Tidak | Normal |
| 99 | l | 33 | 90 | 118 | 199 | 66 | 66 | Tidak | Ya | Ya | Normal |
| 100 | l | 33 | 89 | 93 | 194 | 177 | 129 | Tidak | Tidak | Tidak | Normal |
| 101 | l | 34 | 60 | 99 | 188 | 81 | 71 | Tidak | Ya | Ya | Normal |
| 102 | l | 35 | 85 | 114 | 193 | 115 | 62 | Ya | Ya | Ya | Normal |
| 103 | l | 37 | 99 | 117 | 182 | 180 | 82 | Tidak | Tidak | Tidak | Normal |
| 104 | p | 37 | 85 | 100 | 196 | 74 | 89 | Tidak | Tidak | Tidak | Normal |
| 105 | p | 39 | 87 | 103 | 188 | 72 | 57 | Tidak | Tidak | Tidak | Normal |
| 106 | p | 41 | 95 | 106 | 192 | 171 | 118 | Tidak | Tidak | Ya | Normal |
| 107 | p | 41 | 78 | 117 | 196 | 101 | 129 | Ya | Tidak | Ya | Normal |
| 108 | p | 42 | 74 | 92 | 189 | 102 | 95 | Ya | Ya | Ya | Normal |
| 109 | l | 42 | 93 | 103 | 192 | 156 | 141 | Ya | Ya | Tidak | Normal |
| 110 | p | 42 | 91 | 115 | 184 | 144 | 54 | Tidak | Ya | Tidak | Normal |
| 111 | l | 43 | 83 | 96 | 199 | 103 | 94 | Tidak | Tidak | Ya | Normal |
| 112 | l | 44 | 70 | 108 | 191 | 185 | 130 | Tidak | Tidak | Ya | Normal |
| 113 | p | 44 | 64 | 117 | 194 | 133 | 72 | Tidak | Ya | Tidak | Normal |
| 114 | p | 46 | 70 | 116 | 198 | 71 | 125 | Tidak | Ya | Tidak | Normal |
| 115 | p | 46 | 78 | 97 | 186 | 67 | 64 | Tidak | Ya | Tidak | Normal |
| 116 | l | 47 | 93 | 92 | 182 | 137 | 96 | Ya | Tidak | Ya | Normal |
| 117 | p | 48 | 96 | 94 | 192 | 156 | 131 | Tidak | Tidak | Ya | Normal |
| 118 | l | 49 | 95 | 114 | 191 | 188 | 96 | Tidak | Tidak | Tidak | Normal |
| 119 | p | 52 | 65 | 117 | 184 | 143 | 145 | Tidak | Tidak | Tidak | Normal |
| 120 | p | 52 | 99 | 99 | 187 | 82 | 117 | Tidak | Ya | Tidak | Normal |
| 121 | p | 52 | 80 | 96 | 194 | 198 | 111 | Tidak | Ya | Tidak | Normal |
| 122 | p | 52 | 66 | 110 | 197 | 192 | 138 | Tidak | Tidak | Ya | Normal |

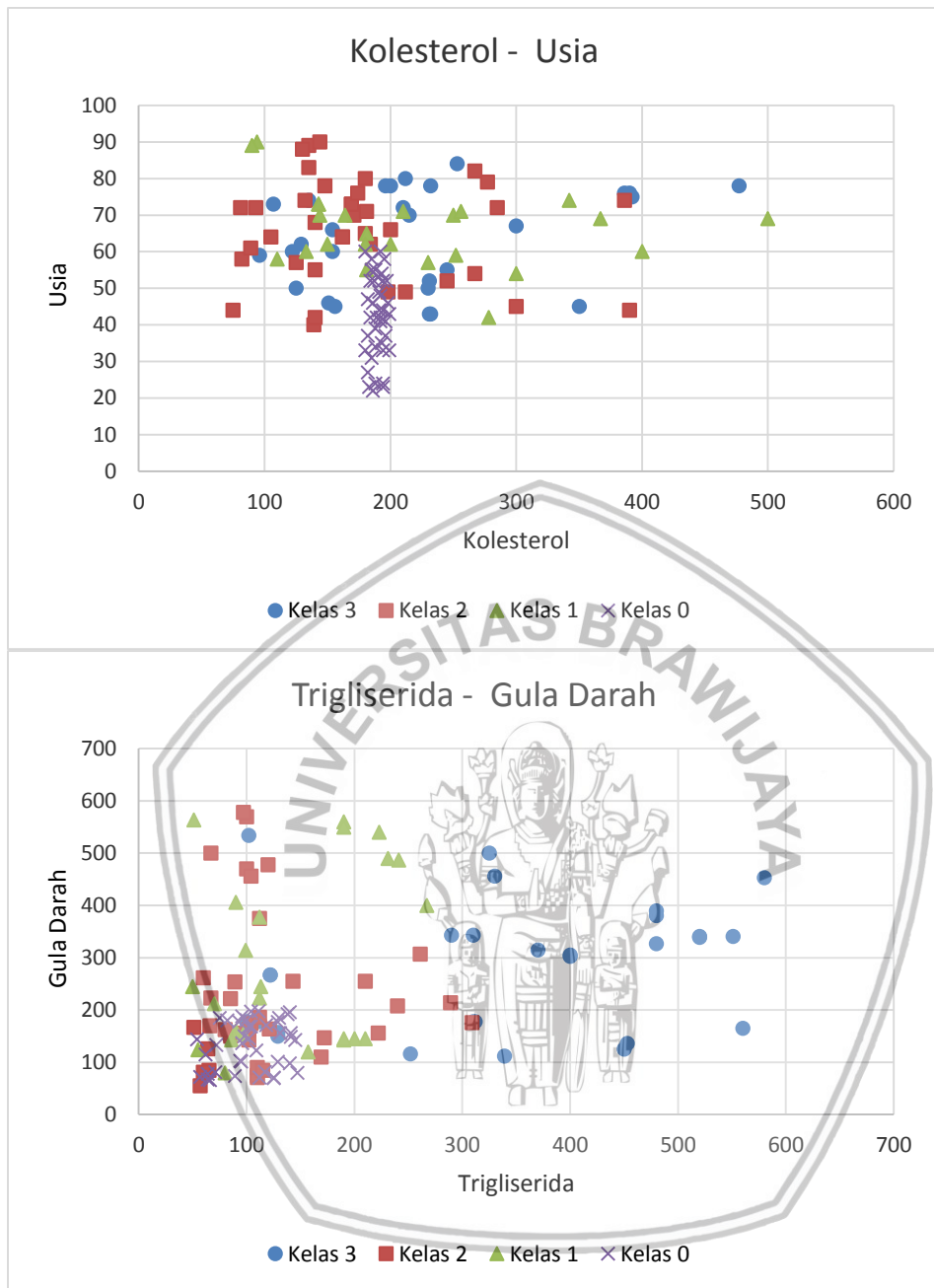
| | | | | | | | | | | | |
|-----|---|----|----|-----|-----|-----|-----|-------|-------|-------|--------|
| 123 | p | 53 | 97 | 98 | 190 | 80 | 147 | Tidak | Tidak | Tidak | Normal |
| 124 | l | 54 | 95 | 111 | 193 | 185 | 75 | Tidak | Tidak | Tidak | Normal |
| 125 | l | 55 | 79 | 111 | 185 | 71 | 112 | Ya | Ya | Tidak | Normal |
| 126 | p | 55 | 84 | 109 | 188 | 78 | 64 | Tidak | Tidak | Tidak | Normal |
| 127 | l | 58 | 67 | 100 | 185 | 149 | 141 | Tidak | Tidak | Tidak | Normal |
| 128 | p | 58 | 98 | 92 | 195 | 169 | 102 | Tidak | Tidak | Ya | Normal |
| 129 | l | 60 | 98 | 104 | 192 | 70 | 125 | Ya | Ya | Tidak | Normal |
| 130 | l | 60 | 88 | 117 | 180 | 197 | 104 | Ya | Tidak | Ya | Normal |

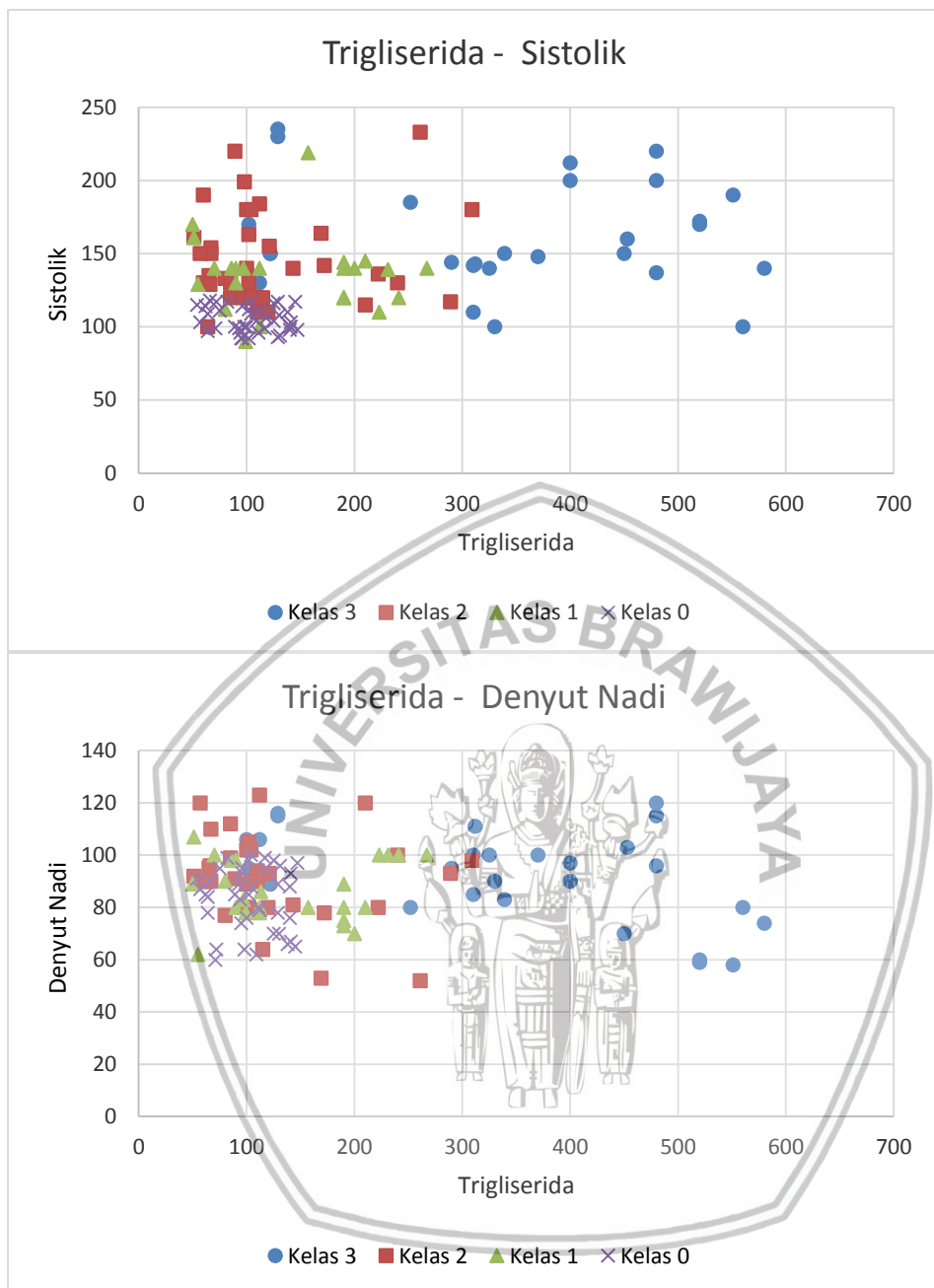


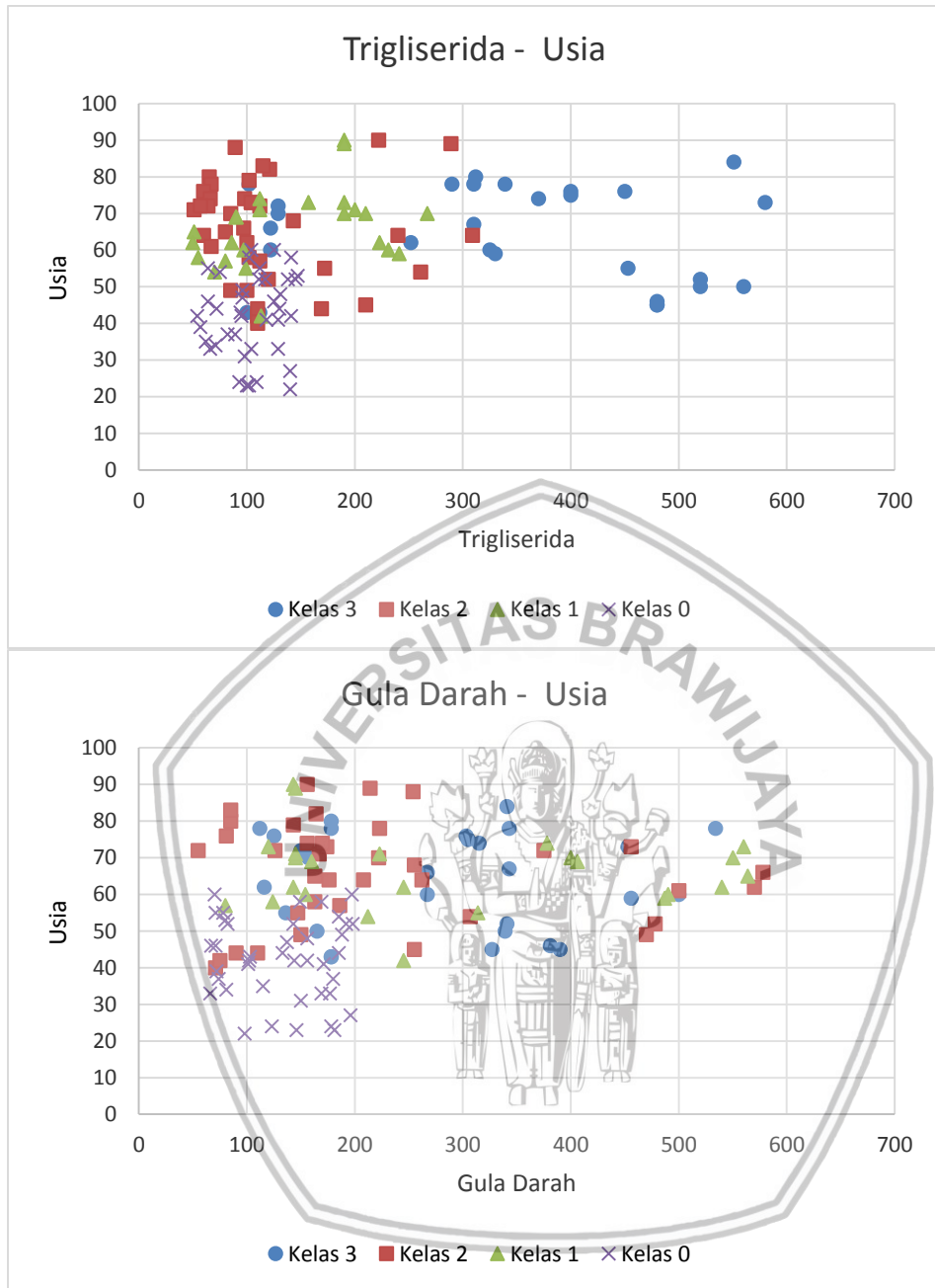
LAMPIRAN B GAMBAR SEBARAN DATA BERAGAI PARAMETER PENYAKIT JANTUNG KORONER

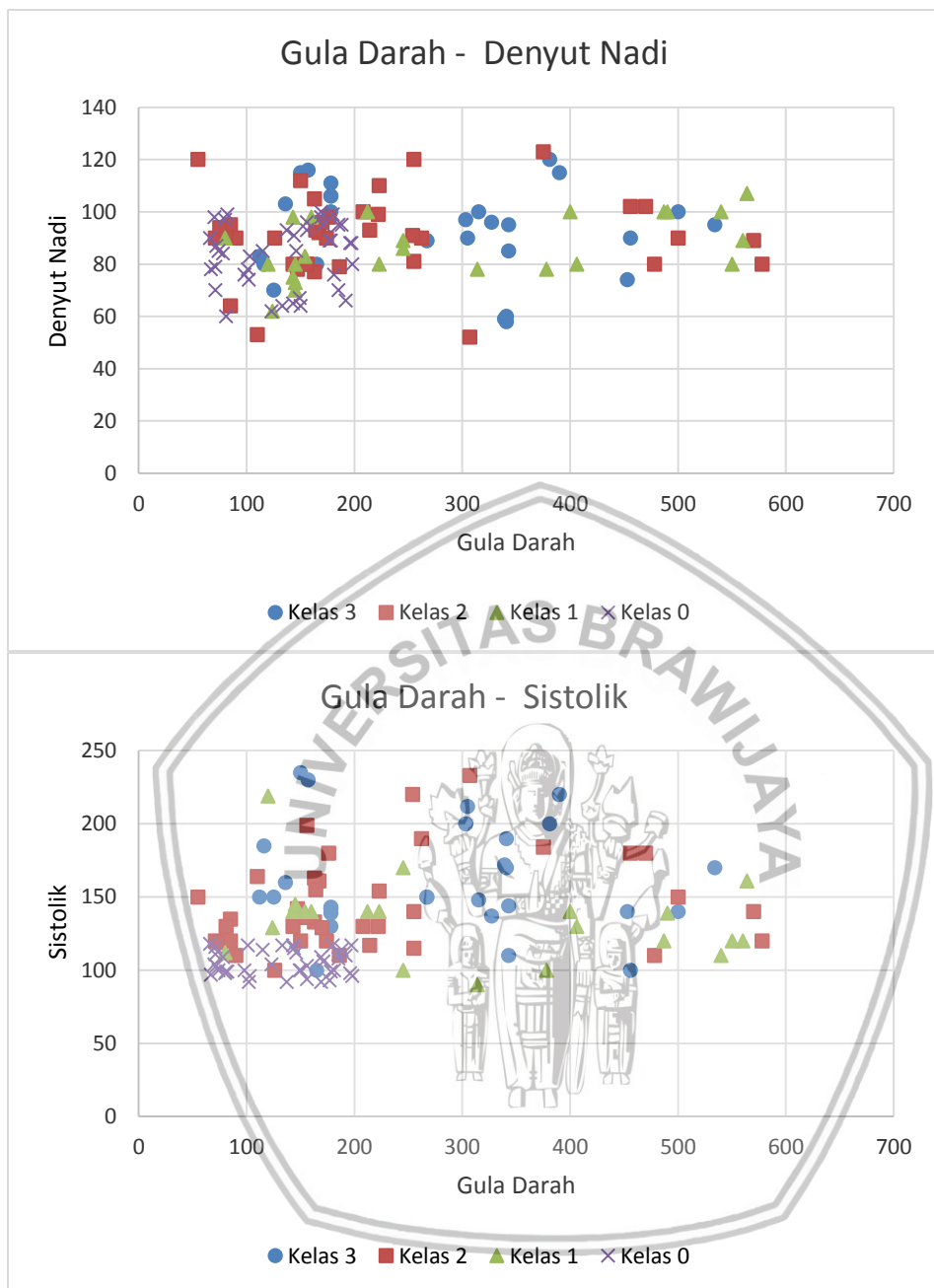


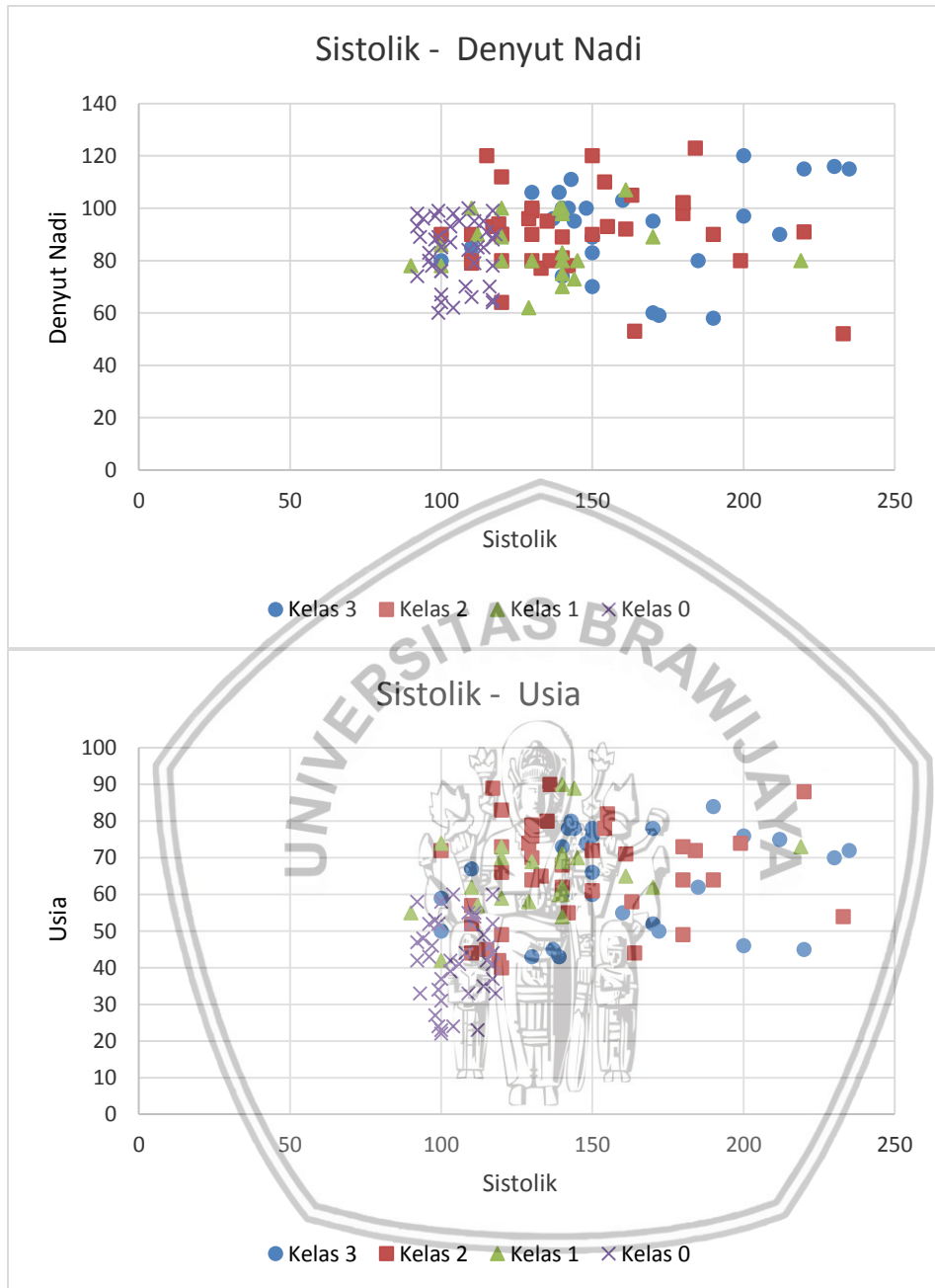


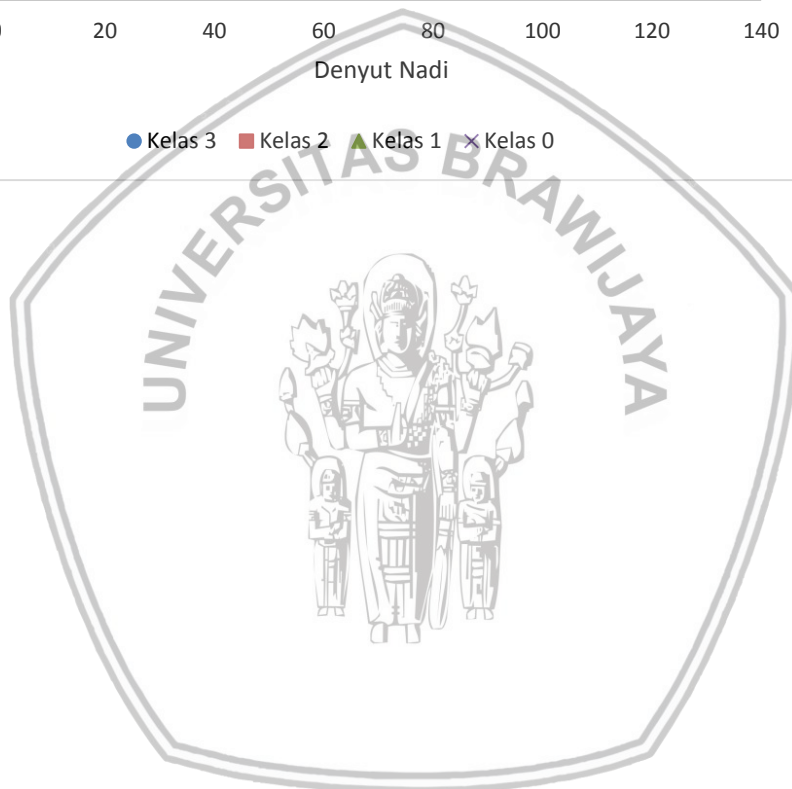
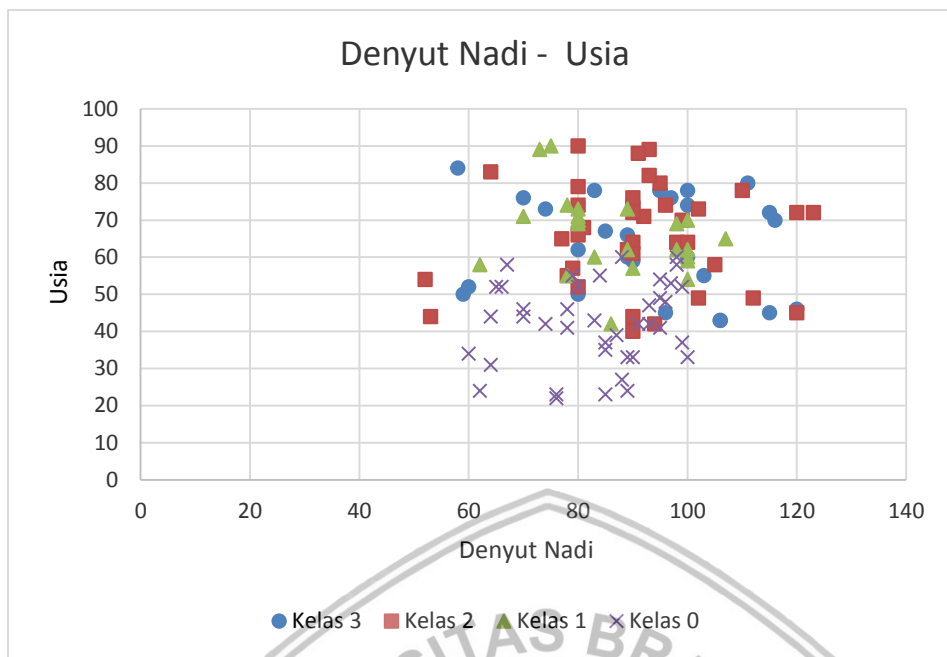












LAMPIRAN C HASIL PENGUJIAN

C.1 Hasil Pengujian Parameter *Learning rate* Untuk Metode LVQ

| Learning rate | Nilai Akurasi Pelatihan dan Pengujian (%) Percobaan ke -i | | | | | | | | | | | | | | | | | | | | Rerata Akurasi | |
|---------------|---|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|----------------|------------|
| | 1 | | 2 | | 3 | | 4 | | 5 | | 6 | | 7 | | 8 | | 9 | | 10 | | Pelatihan | Pengujian |
| | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | | |
| 0,1 | 73,8 46 | 58,4 62 | 73,8 46 | 58,4 62 | 72,3 08 | 58,4 62 | 72,3 08 | 58,4 62 | 73,8 46 | 58,4 62 | 73,8 46 | 58,4 62 | 72,3 08 | 58,4 62 | 73,8 46 | 58,4 62 | 72,3 08 | 58,4 62 | 73,8 46 | 58,4 62 | 73,2 31 | 58,4 62 |
| 0,2 | 55,3 85 | 50,7 69 | 56,9 23 | 53,8 46 | 53,8 46 | 55,3 85 | 41,5 38 | 30,7 69 | 55,3 85 | 52,3 08 | 41,5 38 | 30,7 69 | 63,0 77 | 53,8 46 | 58,4 62 | 52,3 08 | 60,0 00 | 50,7 69 | 55,3 85 | 50,7 69 | 54,1 54 | 48,1 54 |
| 0,3 | 56,9 23 | 53,8 46 | 55,3 85 | 50,7 69 | 41,5 38 | 32,3 08 | 60,0 00 | 52,3 08 | 40,0 00 | 32,3 08 | 60,0 00 | 47,6 92 | 60,0 00 | 52,3 08 | 61,5 38 | 55,3 85 | 58,4 62 | 53,8 46 | 30,7 69 | 30,7 69 | 52,4 62 | 46,1 54 |
| 0,4 | 30,7 69 | 30,7 69 | 29,2 31 | 29,2 31 | 30,7 69 | 30,7 69 | 30,7 69 | 30,7 69 | 21,5 38 | 21,5 38 | 30,7 69 | 30,7 69 | 30,7 69 | 30,7 69 | 29,2 31 | 29,2 31 | 30,7 69 | 30,7 69 | 21,5 38 | 21,5 38 | 28,6 15 | 28,6 15 |
| 0,5 | 21,5 38 | 21,5 38 | 21,5 38 | 21,5 38 | 29,2 31 | 29,2 31 | 18,4 62 | 18,4 62 | 21,5 38 | 21,5 38 | 29,2 31 | 29,2 31 | 29,2 31 | 29,2 31 | 18,4 62 | 18,4 62 | 21,5 38 | 21,5 38 | 29,2 31 | 29,2 31 | 24,0 00 | 24,0 00 |

| Learning rate | Nilai Akurasi Pelatihan dan Pengujian (%) Percobaan ke -i | | | | | | | | | | | | | | | | | | | | Rerata Akurasi | |
|---------------|---|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|----------------|------------|
| | 1 | | 2 | | 3 | | 4 | | 5 | | 6 | | 7 | | 8 | | 9 | | 10 | | Pelatihan | Pengujian |
| | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | | |
| 0,01 | 72,3 08 | 58,4 62 | 72,3 08 | 56,9 23 | 70,7 69 | 56,9 23 | 66,1 54 | 58,4 62 | 70,7 69 | 58,4 62 | 72,3 08 | 58,4 62 | 73,8 46 | 56,9 23 | 72,3 08 | 60,0 00 | 66,1 54 | 58,4 62 | 72,3 08 | 58,4 62 | 70,9 23 | 58,1 54 |
| 0,02 | 70,7 69 | 58,4 62 | 72,3 08 | 58,4 62 | 67,6 92 | 60,0 00 | 70,7 69 | 58,4 62 | 73,8 46 | 56,9 23 | 72,3 08 | 58,4 62 | 76,9 23 | 58,4 62 | 72,3 08 | 58,4 62 | 66,1 54 | 52,3 08 | 73,8 46 | 58,4 62 | 71,6 92 | 57,8 46 |

| | | | | | | | | | | | | | | | | | | | | | | |
|------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
| 0,03 | 66,1 54 | 56,9 23 | 73,8 46 | 58,4 62 | 72,3 08 | 58,4 62 | 73,8 46 | 58,4 62 | 69,2 31 | 61,5 38 | 72,3 08 | 58,4 62 | 72,3 08 | 58,4 62 | 67,6 92 | 60,0 00 | 72,3 08 | 58,4 62 | 67,6 92 | 60,0 00 | 70,7 69 | 58,9 23 |
| 0,04 | 73,8 46 | 58,4 62 | 73,8 46 | 58,4 62 | 73,8 46 | 58,4 62 | 73,8 46 | 58,4 62 | 73,8 46 | 58,4 62 | 72,3 08 | 58,4 62 | 73,8 46 | 58,4 62 | 72,3 08 | 58,4 62 | 72,3 08 | 58,4 62 | 73,8 46 | 58,4 62 | 73,3 85 | 58,4 62 |
| 0,05 | 73,8 46 | 58,4 62 | 72,3 08 | 58,4 62 | 73,8 46 | 58,4 62 | 72,3 08 | 58,4 62 | 72,3 08 | 58,4 62 | 73,8 46 | 58,4 62 | 67,6 92 | 61,5 38 | 72,3 08 | 58,4 62 | 72,3 08 | 58,4 62 | 72,3 08 | 58,4 62 | 72,3 08 | 58,7 69 |
| 0,06 | 73,8 46 | 58,4 62 | 73,8 46 | 58,4 62 | 72,3 08 | 58,4 62 | 73,8 46 | 58,4 62 | 73,8 46 | 58,4 62 | 72,3 08 | 58,4 62 | 73,8 46 | 58,4 62 | 73,8 46 | 58,4 62 | 72,3 08 | 58,4 62 | 73,8 46 | 58,4 62 | 73,3 85 | 58,4 62 |
| 0,07 | 73,8 46 | 58,4 62 | 73,8 46 | 58,4 62 | 72,3 08 | 58,4 62 | 73,8 46 | 58,4 62 | 73,8 46 | 58,4 62 | 72,3 08 | 58,4 62 | 73,8 46 | 58,4 62 | 73,8 46 | 58,4 62 | 73,8 46 | 58,4 62 | 73,8 46 | 58,4 62 | 73,5 38 | 58,4 62 |
| 0,08 | 72,3 08 | 58,4 62 | 73,8 46 | 58,4 62 | 72,3 08 | 58,4 62 | 73,8 46 | 58,4 62 | 73,8 46 | 58,4 62 | 73,8 46 | 58,4 62 | 73,8 46 | 58,4 62 | 73,8 46 | 58,4 62 | 73,8 46 | 58,4 62 | 73,8 46 | 58,4 62 | 73,5 38 | 58,4 62 |
| 0,09 | 73,8 46 | 58,4 62 | 73,8 46 | 58,4 62 | 73,8 46 | 58,4 62 | 73,8 46 | 58,4 62 | 73,8 46 | 58,4 62 | 73,8 46 | 58,4 62 | 73,8 46 | 58,4 62 | 73,8 46 | 58,4 62 | 73,8 46 | 58,4 62 | 73,8 46 | 58,4 62 | 73,8 46 | 58,4 62 |
| 0,1 | 73,8 46 | 58,4 62 | 73,8 46 | 58,4 62 | 73,8 46 | 58,4 62 | 72,3 08 | 58,4 62 | 72,3 08 | 58,4 62 | 72,3 08 | 58,4 62 | 73,8 46 | 58,4 62 | 72,3 08 | 58,4 62 | 73,8 46 | 58,4 62 | 72,3 08 | 58,4 62 | 73,0 77 | 58,4 62 |

C.2 Hasil Pengujian Parameter *Learning rate* Untuk Metode *Backpropagation*

| Learning rate | Nilai Akurasi Pelatihan dan Pengujian (%) Percobaan ke -i | | | | | | | | | | | | | | | | | | | | Rerata Akurasi | |
|---------------|---|--------------|--------------|--------------|-----------|--------------|-----------|--------------|-----------|--------------|--------------|--------------|-----------|--------------|--------------|--------------|-----------|--------------|-----------|--------------|----------------|--------------|
| | 1 | | 2 | | 3 | | 4 | | 5 | | 6 | | 7 | | 8 | | 9 | | 10 | | Pelatihan | Pengujian |
| | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | | |
| 0,1 | 100 | 49,2 3077 | 100 | 50,7 6923 | 100 | 46,1 5385 | 100 | 52,3 0769 | 100 | 50,7 6923 | 100 | 49,2 3077 | 100 | 52,3 0769 | 98,4 6154 | 58,4 6154 | 100 | 50,7 6923 | 100 | 61,5 3846 | 99,8 4615 | 52,1 5385 |
| 0,2 | 100 | 56,9 2308 | 100 | 47,6 9231 | 100 | 56,9 2308 | 100 | 41,5 3846 | 100 | 53,8 4615 | 100 | 36,9 2308 | 100 | 47,6 9231 | 78,4 6154 | 50,7 6923 | 100 | 53,8 4615 | 100 | 60 60 | 97,8 4615 | 50,6 1538 |
| 0,3 | 84,6 1538 | 55,3 8462 | 67,6 9231 | 33,8 4615 | 100 | 52,3 0769 | 100 | 47,6 9231 | 100 | 50,7 6923 | 49,2 3077 | 41,5 3846 | 100 | 58,4 6154 | 58,4 6154 | 52,3 0769 | 100 | 52,3 0769 | 100 | 43,0 7692 | 86 86 | 48,7 6923 |

| | | | | | | | | | | | | | | | | | | | | | | |
|-----|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|-----|--------------|-----|--------------|--------------|--------------|-----|--------------|-----|--------------|--------------|--------------|--------------|--------------|
| 0,4 | 76,9 2308 | 50,7 6923 | 100 | 53,8 4615 | 95,3 8462 | 50,7 6923 | 95,3 8462 | 53,8 4615 | 100 | 52,3 0769 | 100 | 36,9 2308 | 81,5 3846 | 47,6 9231 | 100 | 44,6 1538 | 100 | 52,3 0769 | 93,8 4615 | 53,8 4615 | 94,3 0769 | 49,6 9231 |
| 0,5 | 100 | 63,0 7692 | 64,6 1538 | 38,4 6154 | 92,3 0769 | 43,0 7692 | 64,6 1538 | 44,6 1538 | 100 | 52,3 0769 | 100 | 53,8 4615 | 100 | 53,8 4615 | 100 | 50,7 6923 | 100 | 43,0 7692 | 100 | 41,5 3846 | 92,1 5385 | 48,4 6154 |

| Learning rate | Nilai Akurasi Pelatihan dan Pengujian (%) Percobaan ke -i | | | | | | | | | | | | | | | | | | | | Rerata Akurasi | |
|---------------|---|--------------|--------------|--------------|--------------|--------------|--------------|--------------|-----------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|-----------|--------------|--------------|--------------|----------------|--------------|
| | 1 | | 2 | | 3 | | 4 | | 5 | | 6 | | 7 | | 8 | | 9 | | 10 | | Pelatihan | Pengujian |
| | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | | |
| 0,01 | 100 | 49,2 3077 | 100 | 44,6 1538 | 100 | 50,7 6923 | 90,7 6923 | 49,2 3077 | 100 | 41,5 3846 | 100 | 58,4 6154 | 92,3 0769 | 44,6 1538 | 100 | 46,1 5385 | 100 | 47,6 9231 | 81,5 3846 | 41,5 3846 | 96,4 6154 | 47,3 8462 |
| 0,02 | 100 | 46,1 5385 | 100 | 43,0 7692 | 100 | 29,2 3077 | 100 | 41,5 3846 | 100 | 52,3 0769 | 100 | 46,1 5385 | 100 | 41,5 3846 | 100 | 47,6 9231 | 100 | 52,3 0769 | 100 | 49,2 3077 | 100 | 44,9 2308 |
| 0,03 | 100 | 56,9 2308 | 100 | 49,2 3077 | 86,1 5385 | 53,8 4615 | 100 | 60 100 | 100 | 50,7 6923 | 100 | 50,7 6923 | 100 | 40 100 | 100 | 58,4 6154 | 100 | 44,6 1538 | 100 | 47,6 9231 | 98,6 1538 | 51,2 3077 |
| 0,04 | 100 | 53,8 4615 | 100 | 49,2 3077 | 100 | 53,8 4615 | 100 | 52,3 0769 | 100 | 43,0 7692 | 100 | 52,3 0769 | 100 | 55,3 8462 | 100 | 60 100 | 100 | 56,9 2308 | 100 | 56,9 2308 | 100 | 53,3 8462 |
| 0,05 | 100 | 58,4 6154 | 100 | 56,9 2308 | 87,6 9231 | 35,3 8462 | 100 | 50,7 6923 | 100 | 50,7 6923 | 100 | 53,8 4615 | 100 | 46,1 5385 | 100 | 47,6 9231 | 100 | 56,9 2308 | 100 | 53,8 4615 | 98,7 6923 | 51,0 7692 |
| 0,06 | 100 | 52,3 0769 | 100 | 50,7 6923 | 100 | 47,6 9231 | 100 | 55,3 8462 | 100 | 56,9 2308 | 98,4 6154 | 47,6 9231 | 100 | 55,3 8462 | 100 | 53,8 4615 | 100 | 46,1 5385 | 100 | 49,2 3077 | 99,8 4615 | 51,5 3846 |
| 0,07 | 100 | 52,3 0769 | 98,4 6154 | 52,3 0769 | 100 | 52,3 0769 | 100 | 47,6 9231 | 100 | 52,3 0769 | 100 | 60 100 | 100 | 52,3 0769 | 100 | 47,6 9231 | 100 | 47,6 9231 | 100 | 53,8 4615 | 99,8 4615 | 51,8 4615 |
| 0,08 | 100 | 46,1 5385 | 100 | 55,3 8462 | 100 | 49,2 3077 | 100 | 50,7 6923 | 100 | 53,8 4615 | 96,9 2308 | 52,3 0769 | 100 | 52,3 0769 | 84,6 1538 | 52,3 0769 | 100 | 55,3 8462 | 100 | 53,8 4615 | 98,1 5385 | 52,1 5385 |
| 0,09 | 100 | 49,2 3077 | 100 | 52,3 0769 | 100 | 46,1 5385 | 100 | 50,7 6923 | 100 | 52,3 0769 | 100 | 53,8 4615 | 100 | 49,2 3077 | 100 | 53,8 4615 | 100 | 55,3 8462 | 100 | 44,6 1538 | 100 | 50,7 6923 |
| 0,1 | 100 | 58,4 6154 | 100 | 52,3 0769 | 100 | 58,4 6154 | 100 | 47,6 9231 | 100 | 50,7 6923 | 100 | 58,4 6154 | 100 | 52,3 0769 | 100 | 52,3 0769 | 100 | 53,8 4615 | 98,4 6154 | 52,3 0769 | 99,8 4615 | 53,6 9231 |

C.3 Hasil Pengujian Parameter Pengurangan *Learning rate* Untuk Metode LVQ

| Pengurangan <i>Learning rate</i> | Nilai Akurasi Pelatihan dan Pengujian (%) Percobaan ke -i | | | | | | | | | | | | | | | | | | | | Rerata Akurasi | |
|----------------------------------|---|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|----------------|--------------|
| | 1 | | 2 | | 3 | | 4 | | 5 | | 6 | | 7 | | 8 | | 9 | | 10 | | Pelatihan | Pengujian |
| | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | | |
| 0,01 | 73,8 4615 | 58,4 6154 | 72,3 0769 | 58,4 6154 | 73,8 4615 | 58,4 6154 | 72,3 0769 | 58,4 6154 | 67,6 9231 | 61,5 3846 | 64,6 1538 | 55,3 8462 | 73,8 4615 | 58,4 6154 | 67,6 9231 | 60 | 73,8 4615 | 58,4 6154 | 72,3 0769 | 58,4 6154 | 71,2 3077 | 58,6 1538 |
| 0,02 | 72,3 0769 | 60 | 66,1 5385 | 60 | 66,1 5385 | 56,9 2308 | 73,8 4615 | 58,4 6154 | 72,3 0769 | 60 | 72,3 0769 | 60 | 73,8 4615 | 58,4 6154 | 66,1 5385 | 50,7 6923 | 73,8 4615 | 58,4 6154 | 72,3 0769 | 56,9 2308 | 70,9 2308 | 58 |
| 0,03 | 69,2 3077 | 58,4 6154 | 70,7 6923 | 58,4 6154 | 70,7 6923 | 56,9 2308 | 67,6 9231 | 61,5 3846 | 66,1 5385 | 52,3 0769 | 66,1 5385 | 58,4 6154 | 70,7 6923 | 58,4 6154 | 67,6 9231 | 58,4 6154 | 66,1 5385 | 55,3 8462 | 72,3 0769 | 58,4 6154 | 68,7 6923 | 57,6 9231 |
| 0,04 | 67,6 9231 | 60 | 66,1 5385 | 52,3 0769 | 72,3 0769 | 60 | 73,8 4615 | 55,3 8462 | 72,3 0769 | 60 | 69,2 3077 | 56,9 2308 | 73,8 4615 | 58,4 6154 | 69,2 3077 | 58,4 6154 | 61,5 3846 | 56,9 2308 | 70,7 6923 | 58,4 6154 | 69,6 9231 | 57,6 9231 |
| 0,05 | 66,1 5385 | 55,3 8462 | 66,1 5385 | 53,8 4615 | 69,2 3077 | 47,6 9231 | 67,6 9231 | 60 | 63,0 7692 | 56,9 2308 | 67,6 9231 | 55,3 8462 | 66,1 5385 | 55,3 8462 | 73,8 4615 | 55,3 8462 | 73,8 4615 | 58,4 6154 | 67,6 9231 | 56,9 2308 | 68,1 5385 | 55,5 3846 |
| 0,06 | 64,6 1538 | 53,8 4615 | 73,8 4615 | 53,8 4615 | 63,0 7692 | 58,4 6154 | 72,3 0769 | 60 | 73,8 4615 | 56,9 2308 | 73,8 4615 | 53,8 4615 | 76,9 2308 | 58,4 6154 | 72,3 0769 | 58,4 6154 | 66,1 5385 | 53,8 4615 | 67,6 9231 | 61,5 3846 | 70,4 6154 | 56,9 2308 |
| 0,07 | 63,0 7692 | 55,3 8462 | 72,3 0769 | 55,3 8462 | 69,2 3077 | 56,9 2308 | 72,3 0769 | 60 | 67,6 9231 | 56,9 2308 | 64,6 1538 | 53,8 4615 | 67,6 9231 | 50,7 6923 | 66,1 5385 | 50,7 6923 | 73,8 4615 | 49,2 3077 | 67,6 9231 | 61,5 3846 | 68,4 6154 | 55,0 7692 |
| 0,08 | 67,6 9231 | 47,6 9231 | 67,6 9231 | 53,8 4615 | 66,1 5385 | 53,8 4615 | 66,1 5385 | 63,0 7692 | 63,0 7692 | 56,9 40 | 55,3 2308 | 64,6 8462 | 50,7 1538 | 69,2 6923 | 55,3 3077 | 64,6 8462 | 53,8 1538 | 75,3 4615 | 56,9 8462 | 66,1 2308 | 53,0 5385 | 7692 |
| 0,09 | 70,7 6923 | 60 | 73,8 4615 | 55,3 8462 | 69,2 3077 | 52,3 0769 | 70,7 6923 | 50,7 6923 | 67,6 9231 | 63,0 7692 | 69,2 3077 | 53,8 4615 | 66,1 5385 | 52,3 0769 | 66,1 5385 | 52,3 0769 | 66,1 5385 | 56,9 2308 | 63,0 7692 | 56,9 2308 | 68,3 0769 | 55,3 8462 |
| 0,1 | 66,1 5385 | 53,8 4615 | 67,6 9231 | 49,2 3077 | 56,9 2308 | 55,3 8462 | 67,6 9231 | 53,8 4615 | 73,8 4615 | 52,3 0769 | 69,2 3077 | 61,5 3846 | 69,2 3077 | 56,9 2308 | 64,6 1538 | 49,2 3077 | 73,8 4615 | 50,7 6923 | 61,5 3846 | 61,5 3846 | 67,0 7692 | 54,4 6154 |

C.4 Hasil Pengujian Jumlah Neuron Untuk Metode *Backpropagation*

| Jumlah Neuron | Nilai Akurasi Pelatihan dan Pengujian (%) Percobaan ke -i | | | | | | | | | | | | | | | | | | | | Rerata Akurasi | |
|---------------|---|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|----------------|-----------|
| | 1 | | 2 | | 3 | | 4 | | 5 | | 6 | | 7 | | 8 | | 9 | | 10 | | Pelatihan | Pengujian |
| | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | | |
| 1 | 52,30769 | 43,07692 | 53,84615 | 43,07692 | 56,92308 | 43,07692 | 52,30769 | 43,07692 | 50,76923 | 43,07692 | 52,30769 | 43,07692 | 53,84615 | 44,61538 | 53,84615 | 44,61538 | 52,30769 | 43,07692 | 53,84615 | 43,07692 | 53,23077 | 43,38462 |
| 2 | 72,30769 | 46,15385 | 73,84615 | 47,69231 | 73,84615 | 47,69231 | 56,92308 | 52,30769 | 73,84615 | 47,69231 | 63,07692 | 44,61538 | 58,46154 | 50,76923 | 63,07692 | 43,07692 | 56,92308 | 49,23077 | 73,84615 | 47,69231 | 66,61538 | 47,69231 |
| 3 | 61,53846 | 47,69231 | 75,38462 | 41,53846 | 73,84615 | 36,92308 | 83,07692 | 47,69231 | 83,07692 | 46,15385 | 60231 | 47,69231 | 83,07692 | 52,30769 | 83,07692 | 47,69231 | 64,61538 | 47,69231 | 83,07692 | 46,15385 | 75,07692 | 46,15385 |
| 4 | 80923 | 30,76923 | 100 | 49,23077 | 100 | 43,07692 | 100 | 49,23077 | 100 | 46,15385 | 92,30769 | 33,84615 | 78,46154 | 41,53846 | 90,76923 | 26,15385 | 87,69231 | 36,92308 | 96,92308 | 44,61538 | 92,61538 | 40,15385 |
| 5 | 100 | 41,53846 | 98,46154 | 30,76923 | 100 | 20 | 72,30769 | 35,38462 | 100 | 40 | 100 | 24,61538 | 100 | 43,07692 | 100 | 46,15385 | 95,38462 | 44,61538 | 100 | 40 | 96,61538 | 36,61538 |
| 6 | 100 | 41,53846 | 100 | 40 | 100 | 46,15385 | 100 | 46,15385 | 100 | 47,69231 | 100 | 41,53846 | 100 | 43,07692 | 100 | 40 | 100 | 49,23077 | 100 | 41,53846 | 100 | 43,69231 |
| 7 | 100 | 40 | 100 | 41,53846 | 100 | 46,15385 | 98,46154 | 41,53846 | 100 | 44,61538 | 100 | 30,76923 | 100 | 35,38462 | 100 | 40 | 98,46154 | 41,53846 | 100 | 36,92308 | 99,69231 | 39,84615 |
| 8 | 100 | 40 | 100 | 46,15385 | 100 | 41,53846 | 100 | 41,53846 | 100 | 40 | 100 | 36,92308 | 100 | 35,38462 | 100 | 47,69231 | 100 | 47,69231 | 100 | 49,23077 | 100 | 42,61538 |
| 9 | 100 | 47,69231 | 100 | 43,07692 | 100 | 40 | 100 | 32,30769 | 100 | 44,61538 | 100 | 36,92308 | 100 | 30,76923 | 100 | 49,23077 | 100 | 52,30769 | 100 | 46,15385 | 100 | 42,30769 |
| 10 | 100 | 53,84615 | 100 | 46,15385 | 100 | 40 | 100 | 49,23077 | 100 | 53,84615 | 100 | 36,92308 | 100 | 43,07692 | 100 | 46,15385 | 100 | 46,15385 | 100 | 43,07692 | 100 | 45,84615 |

C.5 Hasil Pengujian Jumlah *Hidden Layer* Untuk Metode *Backpropagation*

| Jumlah <i>Hidden Layer</i> | Nilai Akurasi Pelatihan dan Pengujian (%) Percobaan ke -i | | | | | | | | | | | | | | | | | | | | Rerata Akurasi | |
|----------------------------|---|--------------|------------|--------------|------------|--------------|------------|--------------|------------|--------------|------------|--------------|------------|--------------|------------|--------------|------------|--------------|------------|--------------|----------------|--------------|
| | 1 | | 2 | | 3 | | 4 | | 5 | | 6 | | 7 | | 8 | | 9 | | 10 | | Pelat ihan | Peng ujian |
| | Pelat ihan | Peng ujian | Pelat ihan | Peng ujian | Pelat ihan | Peng ujian | Pelat ihan | Peng ujian | Pelat ihan | Peng ujian | Pelat ihan | Peng ujian | Pelat ihan | Peng ujian | Pelat ihan | Peng ujian | Pelat ihan | Peng ujian | Pelat ihan | Peng ujian | | |
| 1 (10) | 100 | 53,8 4615 | 100 | 46,1 5385 | 100 | 40 | 100 | 49,2 3077 | 100 | 53,8 4615 | 100 | 36,9 2308 | 100 | 43,0 7692 | 100 | 46,1 5385 | 100 | 46,1 5385 | 100 | 43,0 7692 | 100 | 45,8 4615 |
| 2 (10 10) | 100 | 53,8 4615 | 100 | 49,2 3077 | 100 | 49,2 3077 | 100 | 36,9 2308 | 100 | 47,6 9231 | 100 | 53,8 4615 | 100 | 53,8 4615 | 100 | 49,2 3077 | 100 | 60 60 | 100 | 40 40 | 100 | 49,3 8462 |
| 3 (10 10 10) | 100 | 47,6 9231 | 100 | 61,5 3846 | 100 | 63,0 7692 | 100 | 50,7 6923 | 100 | 49,2 3077 | 100 | 53,8 4615 | 100 | 55,3 8462 | 100 | 52,3 0769 | 100 | 55,3 8462 | 100 | 55,3 8462 | 100 | 54,4 6154 |
| 4 (10 10 10 10) | 100 | 56,9 2308 | 100 | 58,4 6154 | 100 | 46,1 5385 | 100 | 56,9 2308 | 100 | 53,8 4615 | 100 | 50,7 6923 | 100 | 50,7 6923 | 100 | 58,4 6154 | 100 | 60 60 | 100 | 41,5 3846 | 100 | 53,3 8462 |

| Jumlah <i>Hidden Layer</i> | Lama Waktu Pelatihan (s) Percobaan ke -i | | | | | | | | | | Rerata Lama Waktu (s) |
|----------------------------|--|--------|--------|--------|--------|--------|--------|--------|--------|--------|-----------------------|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | |
| 1 (10) | 10,253 | 8,136 | 8,736 | 9,985 | 8,727 | 9,341 | 8,974 | 14,59 | 9,173 | 11,089 | 9,9004 |
| 2 (10 10) | 20,135 | 17,855 | 21,168 | 24,822 | 17,417 | 19,737 | 22,142 | 24,558 | 17,805 | 19,648 | 20,5287 |
| 3 (10 10 10) | 31,507 | 26,283 | 28,089 | 22,338 | 25,911 | 27,837 | 29,081 | 21,34 | 27,155 | 31,638 | 27,1179 |
| 4 (10 10 10 10) | 60,606 | 55,338 | 44,757 | 51,634 | 44,111 | 57,468 | 43,65 | 52,164 | 45,924 | 47,951 | 50,3603 |

C.6 Hasil Pengujian Arsitektur Metode *Backpropagation*

| Arsitektur Hidden Layer | Nilai Akurasi Pelatihan dan Pengujian (%) Percobaan ke -i | | | | | | | | | | | | | | | | | | | | Rerata Akurasi | |
|-------------------------------|---|--------------|-----------|--------------|-----------|--------------|-----------|--------------|-----------|--------------|-----------|--------------|-----------|--------------|-----------|--------------|-----------|--------------|-----------|--------------|----------------|--------------|
| | 1 | | 2 | | 3 | | 4 | | 5 | | 6 | | 7 | | 8 | | 9 | | 10 | | Pelatihan | Pengujian |
| | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | | |
| 10 5 2 | 100 | 56,92 308 | 100 | 49,23 077 | 100 | 61,53 846 | 100 | 61,53 846 | 100 | 50,76 923 | 100 | 50,76 923 | 100 | 49,23 077 | 100 | 63,07 692 | 100 | 56,92 308 | 100 | 56,92 308 | 100 | 55,69 231 |
| 10 5 3 | 100 | 52,30 769 | 100 | 50,76 923 | 100 | 60 | 100 | 50,76 923 | 100 | 50,76 923 | 100 | 44,61 538 | 100 | 64,61 538 | 100 | 53,84 615 | 100 | 63,07 692 | 100 | 63,07 692 | 100 | 55,38 462 |
| 10 5 4 | 100 | 53,84 615 | 100 | 53,84 615 | 100 | 56,92 308 | 100 | 49,23 077 | 100 | 52,30 769 | 100 | 56,92 308 | 100 | 53,84 615 | 100 | 58,46 154 | 100 | 56,92 308 | 100 | 58,46 154 | 100 | 55,07 692 |
| 10 5 5 | 100 | 56,92 308 | 100 | 52,30 769 | 100 | 55,38 462 | 100 | 46,15 385 | 100 | 61,53 846 | 100 | 52,30 769 | 100 | 58,46 154 | 100 | 56,92 308 | 100 | 55,38 462 | 100 | 49,23 077 | 100 | 54,46 154 |
| 10 2 5 | 100 | 63,07 692 | 100 | 64,61 538 | 100 | 58,46 154 | 100 | 64,61 538 | 100 | 52,30 769 | 100 | 55,38 462 | 100 | 63,07 692 | 100 | 56,92 308 | 100 | 60 | 100 | 50,76 923 | 100 | 58,92 308 |
| 10 3 5 | 100 | 53,84 615 | 100 | 50,76 923 | 100 | 56,92 308 | 100 | 53,84 615 | 100 | 66,15 385 | 100 | 69,23 077 | 100 | 55,38 462 | 100 | 64,61 538 | 100 | 61,53 846 | 100 | 47,69 231 | 100 | 58 |
| 10 4 5 | 100 | 60 | 100 | 50,76 923 | 100 | 56,92 308 | 100 | 56,92 308 | 100 | 53,84 615 | 100 | 60 | 100 | 63,07 692 | 100 | 56,92 308 | 100 | 50,76 923 | 100 | 52,30 769 | 100 | 56,15 385 |

| Jumlah Hidden Layer | Lama Waktu Pelatihan (s) Percobaan ke -i | | | | | | | | | | Rerata Lama Waktu (s) |
|------------------------|--|--------|--------|--------|--------|--------|--------|--------|--------|--------|-----------------------------|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | |
| 10 5 2 | 15,314 | 17,669 | 20,982 | 16,411 | 13,553 | 16,079 | 17,872 | 15,929 | 14,611 | 12,619 | 16,1039 |
| 10 5 3 | 22,208 | 23,941 | 16,099 | 24,631 | 25,046 | 20,349 | 16,903 | 15,457 | 17,824 | 18,305 | 20,0763 |
| 10 5 4 | 24,772 | 23,312 | 14,863 | 17,372 | 19,918 | 23,011 | 21,559 | 18,275 | 44,911 | 18,19 | 22,6183 |

| | | | | | | | | | | | |
|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|----------------|
| 10 5 5 | 30,658 | 26,427 | 19,337 | 20,748 | 18,585 | 21,015 | 20,212 | 19,097 | 17,245 | 20,829 | 21,4153 |
| 10 2 5 | 13,887 | 13,991 | 11,658 | 13,888 | 13,467 | 12,177 | 13,735 | 15,023 | 14,144 | 11,105 | 13,3075 |
| 10 3 5 | 16,57 | 17,905 | 18,672 | 15,829 | 13,241 | 18,977 | 14,203 | 17,068 | 16,012 | 17,24 | 16,5717 |
| 10 4 5 | 18,135 | 13,947 | 18,496 | 15,797 | 13,464 | 14,217 | 16,769 | 14,773 | 19,074 | 15,204 | 15,9876 |

C.7 Hasil Pengujian Jenis Fungsi Aktivasi Untuk Metode *Backpropagation*

| Jenis Fungsi Aktivasi | Arsitektur Hidden Layer | Nilai Akurasi Pelatihan dan Pengujian (%) Percobaan ke -i | | | | | | | | | | | | | | | | | | | | Rerata Akurasi | |
|-----------------------|-------------------------|---|--------------|--------------|--------------|--------------|--------------|-----------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|-----------|--------------|-----------|--------------|--------------|--------------|----------------|--------------|
| | | 1 | | 2 | | 3 | | 4 | | 5 | | 6 | | 7 | | 8 | | 9 | | 10 | | Pelatihan | Pengujian |
| | | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | | |
| 0 | 10 2 5 | | | | 53,84 615 | | 50,76 923 | | 52,30 769 | | 47,69 231 | | 55,38 462 | 98,4 6154 | 53,84 615 | | 56,92 308 | | 56,92 308 | | 67,69 231 | 99,84 615 | 55,53 846 |
| | 10 3 5 | | 47,69 231 | | 55,38 462 | | 67,69 231 | | 63,07 692 | | 53,84 615 | | 50,76 923 | | 55,38 462 | | 60 | | 60 | | 52,30 769 | | 56,61 538 |
| | 10 4 5 | | 50,76 923 | | 58,46 154 | | 46,15 385 | | 60 | | 61,53 846 | | 56,92 308 | | 52,30 769 | | 56,92 308 | | 53,84 615 | | 53,84 615 | | 55,07 692 |
| 1 | 10 2 5 | 92,3 0769 | 55,38 462 | 90,7 6923 | 44,61 538 | 75,3 8462 | 52,30 769 | | 60 | 87,6 9231 | 47,69 231 | 89,2 3077 | 49,23 077 | | 64,61 538 | 80 | 53,84 615 | 100 | 49,23 077 | 92,3 0769 | 55,38 462 | 90,76 923 | 53,23 077 |
| | 10 3 5 | | 44,61 538 | | 61,53 846 | | 46,15 385 | | 55,38 462 | | 56,92 308 | | 55,38 462 | | 58,46 154 | | 52,30 769 | | 58,46 154 | | | | 54,92 308 |
| | 10 4 5 | | 60 | | 52,30 769 | | 60 | | 58,46 154 | | 53,84 615 | | 47,69 231 | | 47,69 231 | | 55,38 462 | | 60 | | 61,53 846 | | 55,69 231 |

| Arsitektur Hidden Layer | Lama Waktu Pelatihan (s) Percobaan ke -i | | | | | | | | | | Rerata Lama Waktu (s) |
|----------------------------|--|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------------------------------|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | |
| 10 2 5 | 14,633 | 9,585 | 12,408 | 13,083 | 13,047 | 12,005 | 74,6 | 17,102 | 13,749 | 14,034 | 19,4246 |
| 10 3 5 | 22,835 | 19,224 | 16,814 | 14,995 | 19,266 | 19,689 | 21,158 | 16,971 | 12,676 | 16,944 | 18,0572 |
| 10 4 5 | 24,005 | 17,187 | 19,115 | 18,824 | 18,39 | 19,458 | 23,153 | 19,987 | 14,671 | 13,915 | 18,8705 |
| 10 2 5 | 91,504 | 59,523 | 54,804 | 5,236 | 55,347 | 55,135 | 20,161 | 55,526 | 5,037 | 54,24 | 45,6513 |
| 10 3 5 | 8,01 | 9,468 | 8,908 | 7,009 | 7,812 | 5,989 | 9,142 | 7,202 | 7,227 | 7,345 | 7,8112 |
| 10 4 5 | 11,878 | 8,669 | 8,808 | 8,048 | 8,577 | 8,034 | 7,097 | 7,543 | 8,198 | 6,23 | 8,3082 |

C.8 Hasil Pengujian Pasangan Data Latih dan Uji Untuk Metode LVQ

| Pasangan Data Latih dan Uji | Nilai Akurasi Pelatihan dan Pengujian (%) Percobaan ke -i | | | | | | | | | | | | | | | | | | | | Rerata Akurasi | |
|--------------------------------|---|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|-------------------|--------------|
| | 1 | | 2 | | 3 | | 4 | | 5 | | 6 | | 7 | | 8 | | 9 | | 10 | | Pelatihan | Pengujian |
| | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | | |
| (65,65) / 50:50 | 73,84 6154 | 58,46 1538 | 73,84 6154 | 58,46 1538 | 66,15 3846 | 56,92 3077 | 72,30 7692 | 58,46 1538 | 73,84 6154 | 58,46 1538 | 66,15 3846 | 60 | 72,30 7692 | 58,46 1538 | 73,84 6154 | 58,46 1538 | 67,69 2308 | 60 | 73,84 6154 | 58,46 1538 | 71,3 8462 | 58,6 1538 |
| (78,52) / 60:40 | 70,51 2821 | 42,30 7692 | 71,79 4872 | 42,30 7692 | 76,92 3077 | 48,07 6923 | 71,79 4872 | 42,30 7692 | 74,35 8974 | 46,15 3846 | 76,92 3077 | 46,15 3846 | 57,69 2308 | 42,30 7692 | 78,20 5128 | 48,07 6923 | 76,92 3077 | 46,15 3846 | 74,35 8974 | 46,15 3846 | 72,9 4872 | 45 |
| (92,38) / 70:30 | 69,56 5217 | 52,63 1579 | 69,56 5217 | 55,26 3158 | 68,47 8261 | 52,63 1579 | 70,65 2174 | 55,26 3158 | 68,47 8261 | 52,63 1579 | 65,21 7391 | 44,73 6842 | 67,39 1304 | 52,63 1579 | 69,56 5217 | 52,63 1579 | 59,78 2609 | 42,10 5263 | 65,21 7391 | 44,73 6842 | 67,3 913 | 50,5 2632 |
| (103,27) / 80:20 | 70,87 3786 | 59,25 9259 | 70,87 3786 | 59,25 9259 | 70,87 3786 | 59,25 9259 | 66,99 0291 | 70,37 037 | 70,87 3786 | 59,25 9259 | 70,87 3786 | 59,25 9259 | 69,90 2913 | 59,25 9259 | 71,84 466 | 66,66 6667 | 73,78 6408 | 74,07 4074 | 69,90 2913 | 55,55 5556 | 70,6 7961 | 62,2 2222 |

| Arsitektur Hidden Layer | Lama Waktu Pelatihan (s) Percobaan ke -i | | | | | | | | | | Rerata Lama Waktu (s) |
|----------------------------|--|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------------------------------|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | |
| (65,65) / 50:50 | 11,533 | 10,552 | 10,663 | 10,615 | 11,002 | 10,661 | 10,559 | 10,447 | 11,008 | 10,55 | 10,759 |
| (78,52) / 60:40 | 12,663 | 14,181 | 17,555 | 20,795 | 23,511 | 23,168 | 22,605 | 25,086 | 22,798 | 24,201 | 20,6563 |
| (92,38) / 70:30 | 20,971 | 24,767 | 26,697 | 25,791 | 25,29 | 24,896 | 25,475 | 20,529 | 18,321 | 19,043 | 23,178 |
| (103,27) / 80:20 | 30,499 | 27,669 | 26,396 | 26,48 | 26,872 | 27,024 | 22,484 | 19,882 | 18,561 | 16,88 | 24,2747 |

C.9 Hasil Pengujian Pasangan Data Latih dan Uji Untuk Metode *Backpropagation*

| Pasangan Data Latih dan Uji | Nilai Akurasi Pelatihan dan Pengujian (%) Percobaan ke -i | | | | | | | | | | | | | | | | | | | | Rerata Akurasi | |
|--------------------------------|---|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|-------------------|---------------|
| | 1 | | 2 | | 3 | | 4 | | 5 | | 6 | | 7 | | 8 | | 9 | | 10 | | Pelat ihan | Peng ujian |
| | Pelat han | Peng ujian | Pelat ihan | Peng ujian | Pelat han | Peng ujian | Pelat ihan | Peng ujian | Pelat ihan | Peng ujian | Pelat ihan | Peng ujian | Pelat ihan | Peng ujian | Pelat ihan | Peng ujian | Pelat ihan | Peng ujian | Pelat ihan | Peng ujian | | |
| (65,65) / 50:50 | 100 | 53,84 6154 | 100 | 52,30 7692 | 100 | 60 | 100 | 53,84 6154 | 100 | 52,30 7692 | 100 | 47,69 2308 | 100 | 52,30 7692 | 100 | 60 | 100 | 63,07 6923 | 100 | 53,84 6154 | 100 | 54,92 308 |
| (78,52) / 60:40 | 100 | 38,46 1538 | 100 | 44,23 0769 | 100 | 42,30 7692 | 100 | 38,46 1538 | 100 | 44,23 0769 | 100 | 36,53 8462 | 100 | 26,92 3077 | 100 | 42,30 7692 | 100 | 42,30 7692 | 100 | 40,38 4615 | 100 | 39,61 538 |
| (92,38) / 70:30 | 98,91 3043 | 52,63 1579 | 100 | 44,73 6842 | 100 | 63,15 7895 | 100 | 42,10 5263 | 100 | 44,73 6842 | 100 | 42,10 5263 | 100 | 28,94 7368 | 100 | 36,84 2105 | 100 | 39,47 3684 | 100 | 44,73 6842 | 99,89 13 | 43,94 737 |
| (103,27) / 80:20 | 100 | 55,55 5556 | 100 | 62,96 2963 | 90,29 1262 | 62,96 2963 | 100 | 74,07 4074 | 100 | 74,07 4074 | 100 | 62,96 2963 | 100 | 66,66 6667 | 100 | 62,96 2963 | 100 | 62,96 2963 | 100 | 59,25 9259 | 99,02 913 | 64,44 444 |

| Pasangan Data Latih dan Uji | Lama Waktu Pelatihan (s) Percobaan ke -i | | | | | | | | | | Rerata Lama Waktu (s) |
|--------------------------------|--|--------|--------|-------|-------|--------|--------|------|--------|-------|-----------------------------|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | |
| (65,65) / 50:50 | 14,969 | 13,004 | 10,263 | 11,67 | 11,58 | 19,245 | 19,384 | 23,5 | 15,393 | 21,03 | 16,0038 |

| | | | | | | | | | | | |
|------------------|---------|--------|--------|--------|--------|--------|--------|--------|--------|--------|---------|
| (78,52) / 60:40 | 30,613 | 48,191 | 55,911 | 26,207 | 31,576 | 30,21 | 23,374 | 23,997 | 25,836 | 29,181 | 32,5096 |
| (92,38) / 70:30 | 126,665 | 43,73 | 57,974 | 22,421 | 19,496 | 48,877 | 17,494 | 49,072 | 36,83 | 23,555 | 44,6114 |
| (103,27) / 80:20 | 70,172 | 77,287 | 94,279 | 22,837 | 24,09 | 32,948 | 75,489 | 17,548 | 30,655 | 24,44 | 46,9745 |

C.10 Hasil Pengujian *Cross Validation* Untuk Metode LVQ

| Pasan gan Data Latih dan Uji | Nilai Akurasi Pelatihan dan Pengujian (%) Percobaan ke -i | | | | | | | | | | | | | | | | | | | | Rerata Akurasi | |
|---|---|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|----------------|---------------|
| | 1 | | 2 | | 3 | | 4 | | 5 | | 6 | | 7 | | 8 | | 9 | | 10 | | Pelati han | Peng ujian |
| | Pelati han | Peng ujian | Pelati han | Peng ujian | Pelati han | Peng ujian | Pelati han | Peng ujian | Pelati han | Peng ujian | Pelati han | Peng ujian | Pelati han | Peng ujian | Pelati han | Peng ujian | Pelati han | Peng ujian | Pelati han | Peng ujian | | |
| 1 (104, 26) | 67,30 769 | 50 | 67,30 769 | 50 | 67,30 769 | 50 | 65,38 462 | 53,84 615 | 66,34 615 | 50 | 66,34 615 | 50 | 60,57 692 | 53,84 615 | 65,38 462 | 53,84 615 | 60,57 692 | 50 | 67,30 769 | 50 | 65,38 462 | 51,15 385 |
| 2 (104, 26) | 64,42 308 | 57,69 231 | 66,34 615 | 61,53 846 | 68,26 923 | 61,53 846 | 55,76 923 | 46,15 385 | 68,26 923 | 61,53 846 | 64,42 308 | 53,84 615 | 54,80 769 | 50 | 64,42 308 | 53,84 615 | 68,26 923 | 61,53 846 | 64,42 308 | 57,69 231 | 63,94 231 | 56,53 846 |
| 3 (104, 26) | 72,11 538 | 65,38 462 | 69,23 077 | 69,23 077 | 72,11 538 | 61,53 846 | 65,38 462 | 61,53 846 | 57,69 231 | 57,69 231 | 69,23 077 | 50 | 64,42 308 | 50 | 69,23 077 | 50 | 64,42 308 | 73,07 692 | 69,23 077 | 53,84 615 | 67,30 769 | 59,23 077 |
| 4 (105, 25) | 68,57 143 | 52 | 63,80 952 | 52 | 64,76 19 | 52 | 66,66 667 | 52 | 67,61 905 | 52 | 63,80 952 | 52 | 66,66 667 | 52 | 64,76 19 | 52 | 64,76 19 | 52 | 66,66 667 | 52 | 65,80 952 | 52 |
| 5 (103, 27) | 71,84 466 | 48,14 815 | 74,75 728 | 55,55 556 | 71,84 466 | 48,14 815 | 73,78 641 | 55,55 556 | 71,84 466 | 48,14 815 | 74,75 728 | 55,55 556 | 73,78 641 | 55,55 556 | 61,16 505 | 48,14 815 | 73,78 641 | 55,55 556 | 72,81 553 | 55,55 556 | 72,03 883 | 52,59 259 |

| Pasangan Data Latih dan Uji | Lama Waktu Pelatihan (s) Percobaan ke -i | | | | | | | | | | Rerata Lama Waktu (s) |
|--------------------------------|--|--------|--------|--------|--------|--------|--------|--------|--------|--------|-----------------------------|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | |
| 1 (104,26) | 18,088 | 17,452 | 16,868 | 17,193 | 17,351 | 17,126 | 17,036 | 17,343 | 17,637 | 17,518 | 17,3612 |
| 2 (104,26) | 18,325 | 17,358 | 17,997 | 17,37 | 17,583 | 17,282 | 17,529 | 17,476 | 17,618 | 17,553 | 17,6091 |
| 3 (104,26) | 18,106 | 17,525 | 17,436 | 17,465 | 17,682 | 17,001 | 17,51 | 17,237 | 17,614 | 17,273 | 17,4849 |
| 4 (105,25) | 18,835 | 17,467 | 17,761 | 17,382 | 17,455 | 17,425 | 17,504 | 17,352 | 17,343 | 17,574 | 17,6098 |
| 5 (103,27) | 18,475 | 17,16 | 17,63 | 17,102 | 17,436 | 17,37 | 17,256 | 17,071 | 17,161 | 17,411 | 17,4072 |

C.11 Hasil Pengujian *Cross Validation* Untuk Metode *Backpropagation*

| Pasangan Data Latih dan Uji | Nilai Akurasi Pelatihan dan Pengujian (%) Percobaan ke -i | | | | | | | | | | | | | | | | | | | | Rerata Akurasi | |
|--------------------------------|---|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|-----------|--------------|--------------|--------------|----------------|--------------|
| | 1 | | 2 | | 3 | | 4 | | 5 | | 6 | | 7 | | 8 | | 9 | | 10 | | Pelatihan | Pengujian |
| | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | Pelatihan | Pengujian | | |
| 1 (104,26) | 100 | 57,69 231 | 100 | 65,38 462 | 100 | 73,07 692 | 81,73 077 | 65,38 462 | 89,42 308 | 80,76 923 | 95,19 231 | 73,07 692 | 78,84 615 | 65,38 462 | 100 | 53,84 615 | 100 | 73,07 692 | 89,42 308 | 69,23 077 | 93,46 154 | 67,69 231 |
| 2 (104,26) | 88,46 154 | 76,92 308 | 100 | 65,38 462 | 100 | 69,23 077 | 69,23 077 | 65,38 462 | 91,34 615 | 76,92 308 | 100 | 84,61 538 | 98,07 692 | 76,92 308 | 100 | 76,92 308 | 100 | 69,23 077 | 97,78 846 | 73,07 692 | | |
| 3 (104,26) | 97,11 538 | 69,23 077 | 100 | 76,92 308 | 100 | 69,23 077 | 97,11 538 | 73,07 692 | 94,23 077 | 80,76 923 | 98,07 692 | 69,23 077 | 73,07 692 | 87,5 | 100 | 69,23 077 | 100 | 80,76 923 | 96,15 385 | 69,23 077 | 97,01 923 | 73,07 692 |
| 4 (105,25) | 98,09 524 | 84 | 97,14 286 | 72 | 97,14 286 | 80 | 100 | 72 | 100 | 72 | 97,14 286 | 72 | 97,14 286 | 72 | 99,04 762 | 76 | 100 | 72 | 96,19 048 | 72 | 98,19 048 | 74,4 |
| 5 (103,27) | 100 | 51,85 185 | 100 | 59,25 926 | 68,93 204 | 48,14 815 | 93,20 388 | 66,66 667 | 100 | 55,55 556 | 100 | 62,96 296 | 72,81 553 | 48,14 815 | 100 | 55,55 556 | 100 | 55,55 556 | 100 | 51,85 185 | 93,49 515 | 55,55 556 |

| Pasangan Data Latih dan Uji | Lama Waktu Pelatihan (s) Percobaan ke -i | | | | | | | | | | <i>Rerata Lama Waktu (s)</i> |
|--------------------------------|--|---------|---------|---------|--------|---------|---------|--------|--------|--------|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | |
| 1 (104,26) | 54,611 | 26,341 | 17,332 | 93,813 | 94,413 | 94,085 | 95,215 | 45,847 | 34,694 | 93,703 | 65,0054 |
| 2 (104,26) | 98,769 | 83,846 | 73,413 | 55,703 | 53,534 | 95,091 | 60,699 | 95,082 | 48,405 | 27,601 | 69,2143 |
| 3 (104,26) | 97,454 | 71,149 | 55,337 | 94,05 | 94,202 | 90,949 | 90,698 | 26,224 | 25,487 | 90,653 | 73,6203 |
| 4 (105,25) | 98,592 | 103,153 | 104,714 | 64,869 | 84,479 | 104,689 | 99,094 | 93,763 | 93,728 | 93,813 | 94,0894 |
| 5 (103,27) | 26,216 | 17,505 | 102,051 | 101,969 | 40,858 | 19,972 | 102,166 | 25,646 | 19,463 | 33,176 | 48,9022 |

